MIT.7238

UNITED STATES PATENT APPLICATION

of

JOSEPH G. DESLOGE

for

LOCATION-ESTIMATING, NULL STEERING (LENS)
ALGORITHM FOR ADAPTIVE ARRAY PROCESSING

LOCATION-ESTIMATING, NULL STEERING (LENS)
ALGORITHM FOR ADAPTIVE ARRAY PROCESSING

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority from the provisional application designated serial number

5   60/100,291 filed September 14, 1998 and entitled "Location-Estimating, Null Steering (LENS)

Algorithm for Adaptive Array Processing".

GOVERNMENT RIGHTS

This invention was made with government support under Contract NIH-5R01-DC00270 by the

10   National Institute of Health. The government may have certain rights in the invention.

BACKGROUND OF THE INVENTION

The present invention relates to the field of digital signal processing, and in particular to the field

to adaptive digital filters for beamforming transducer arrays.

15

Many applications (e.g., radar, sonar, satellite communications, background noise reducing

hearing aids, etc.) often include a directionally sensitive antenna-array or microphone array

(generically a beamforming traducer array). These beamforming transducer arrays and their

associated signal processing are capable of enhancing a desired target source relative to another

20   non-target source (e.g., a jammer/noise source). The enhancements improve the perceived target-

2

to-jammer power ratio (TJR), which, in turn, improves the performance of subsequent, target-based processing.

One application of a beamforming transducer array is for hearing aids. Hearing aid wearers often complain about the corrupting influence of background noise upon their ability to understand a desired target source. Enhancing the target relative to the background noise reduces this problem. Similarly, applications such as satellite communications, radar and sonar also benefit from the directional filtering ability of beamforming transducer arrays. We shall now discuss the basic beamforming problem.

2   Beamforming Background

2.1   Basic Beamforming Problem

The overall beamforming goal is to extract a desired target signal from within a field of non-target, jammer signals using an array of spatially-separated sensors. Specifically, beamformers exploit input differences arising from different source directions of arrival to create a 'spatial-filter' that preserves the target source while nulling the jammer sources. Figure 1 shows the basic beamforming system structure, which filters (convolves) each array input $x_i[n]$, $i = \{0, 1, \ldots, M-1\}$ with a corresponding impulse response $w_i[n]$ and sums the resulting signals to form the overall output $y[n]$:

$$y[n] = \sum_{i=0}^{M-1} w_i[n] * x_i[n].$$ (2.1)

3

Converting Equation 2.1 into the frequency domain and introducing vector notation allows for the beamforming operation to be expressed as an inner product:

$$Y(\omega) = \sum_{i=0}^{M-1} W_i^*(\omega)X_i(\omega) = \underline{W}^H(\omega)\underline{X}(\omega), \qquad (2.2)$$

where $x_i[n] \overset{\mathcal{F}}{\leftrightarrow} X_i(\omega)$, $w_i[n] \overset{\mathcal{F}}{\leftrightarrow} W_i^*(\omega)$, $y[n] \overset{\mathcal{F}}{\leftrightarrow} Y(\omega)$, $\underline{X}(\omega) = [X_0(\omega), \ldots, X_{M-1}(\omega)]^T$, $\underline{W}(\omega) = [W_0(\omega), \ldots, W_{M-1}(\omega)]^T$, and H is the hermitian (complex-conjugate transpose) operation.[1] The main issue involved in the design of a beamforming system is the selection of the array weights $\underline{W}(\omega)$ that yield the desired spatial-filtering behavior. As described in Section 2.3, these weights may be either time-invariant or time-varying. Time-invariant weights typically achieve simple and robust processing, while time-varying weights are often capable of superior performance.

2.2   Beamforming Terminology

Before discussing the beamforming weight-selection process, this section introduces the basic terminology that is used to discuss beamforming systems.

2.2.1   Coordinate System

Beamforming systems use spatially-separated sensors to extract a target source from within a field of jammer sources. The spatial dimension of this task requires a coordinate system to describe both array element and source locations.

Figure 2 shows the three-dimensional, polar-coordinate triple, $(\rho_0, \theta_0, \phi_0)$, that corresponds to the cartesian coordinate triple (x_0, y_0, z_0). In this representation, (1) ρ_0 is the distance from (x_0, y_0, z_0) to the origin, (2) $\theta_0 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the angle between (x_0, y_0, z_0) and the yz-plane (positive if $x_0 > 0$ and negative if $x_0 < 0$), and (3) $\phi_0 \in [-\pi, \pi]$ is the angle between the projection $(0, y_0, z_0)$ of (x_0, y_0, z_0) into the yz-plane and the positive z-axis. The triples (x_0, y_0, z_0) and $(\rho_0, \theta_0, \phi_0)$ are related as follows:

$$\rho_0 = \sqrt{x_0^2 + y_0^2 + z_0^2}$$
$$\theta_0 = \tan^{-1} \frac{x_0}{\sqrt{y_0^2 + z_0^2}} \qquad (2.3)$$
$$\phi_0 = \tan^{-1} \frac{y_0}{z_0}$$

This polar coordinate definition is slightly different from traditional polar-coordinate definitions, but it simplifies the analysis of broadside arrays (Section 2.2.2), which are used for much of this research.

In the analysis of beamformer behavior, it is assumed that all sources are far-field and produce plane waves that propagate across the array, i.e., $\rho = \infty$, which means that (θ, ϕ) alone specifies source locations. The theoretical target source location is assumed to be known and fixed along the positive z-axis: $(\theta_T, \phi_T) = (0, 0)$.

2.2.2 Array Configurations

The array elements lie at known, fixed locations within the array environment and may be arranged in any desired configuration. One special type of array is a uniform linear array (ULA), in which the array elements are co-linear and equally-spaced by a distance d. Figure 3 shows two ULA examples, which are defined relative to a specified target location (along the positive z-axis). For these two configurations, endfire and broadside, the array elements are respectively oriented coaxial with and perpendicular to the target propagation direction.

The simple and regular structure of ULAs allows for straightforward system anal-ysis, and, for this reason, the specific systems that are designed and simulated in the following chapters will have broadside ULA configurations. The coordinate system of Section 2.2.1 is convenient for the broadside ULAs of Figure 3b, because only the θ location parameter is needed to describe the impact of a specific source upon the free-field array (due to the cylindrically-symmetric behavior about the array axis).

2.2.3 Source-to-Array Transfer Function, $\underline{H}(\omega, \theta, \phi)$

The array element locations, the source locations, and the knowledge of the source propagation rules within the array environment allow for the definition of a source-to-array transfer function that relates each source to each array element. Consider a source, $S(\omega, \theta, \phi)$, located at (θ, ϕ). This source produces a signal $\tilde{H}_i(\omega, \theta, \phi)S(\omega, \theta, \phi)$ at array element i, where $\tilde{H}_i(\omega, \theta, \phi)$ is the source-to-element transfer function. The overall array input due to $S(\omega, \theta, \phi)$ is thus

$$\underline{X}(\omega) = \begin{bmatrix} X_0(\omega) \\ X_1(\omega) \\ \vdots \\ X_{M-1}(\omega) \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{H}_0(\omega, \theta, \phi) \\ \tilde{H}_1(\omega, \theta, \phi) \\ \vdots \\ \tilde{H}_{M-1}(\omega, \theta, \phi) \end{bmatrix}}_{\underline{\tilde{H}}(\omega,\theta,\phi)} S(\omega, \theta, \phi), \tag{2.4}$$

where $\underline{\tilde{H}}(\omega, \theta, \phi)$ is the source-to-array transfer function vector. Because beamforming exploits the relative differences between the input signals, the source-to-array transfer functions are usually normalized by $\underline{\tilde{H}}_0(\omega, \theta, \phi)$. Defining $H_i(\omega, \theta, \phi) = \tilde{H}_i(\omega, \theta, \phi)/\tilde{H}_0(\omega, \theta, \phi)$, the normalized source-to-array transfer function vector is

$$\underline{H}(\omega, \theta, \phi) = \frac{\underline{\tilde{H}}(\omega, \theta, \phi)}{\tilde{H}_0(\omega, \theta, \phi)} = \begin{bmatrix} 1 \\ H_1(\omega, \theta, \phi) \\ \vdots \\ H_{M-1}(\omega, \theta, \phi) \end{bmatrix}. \tag{2.5}$$

Under the assumption of far-field sources that propagate as lossless plane-waves across the array and that the array is operating in the free-field, it is possible to determine $\underline{H}(\omega, \theta, \phi)$ explicitly. In this case, the individual-element transfer functions

$H_i(\omega, \theta, \phi)$ are pure delays, since the amplitude does not decay,

$$H_i(\omega, \theta, \phi) = e^{j\omega\tau_i(\theta,\phi)}.$$

The delay $\tau_i(\theta, \phi)$ is equal to the additional distance, $d_i(\theta, \phi)$, that the propagating wave travels between elements 0 and i, divided by the propagation velocity c:

$$H_i(\omega, \theta, \phi) = e^{j\omega\frac{d_i(\theta,\phi)}{c}}.$$

The array geometry provides a means for determining the $d_i(\theta, \phi)$ for a given source location. Note that, since element 0 is used as the reference, $d_0(\omega, \theta, \phi) = 0$, which yields $H_0(\omega, \theta, \phi) = 1$ as desired in Equation 2.5.

For a free-field ULA, each $d_i(\theta, \phi)$ may be stated in terms of $d_1(\theta, \phi)$. Specifically, the equal inter-element spacing and the linear array orientation result in

$$d_i(\omega, \theta, \phi) - d_{i-1}(\omega, \theta, \phi) = d_1(\omega, \theta, \phi), \quad i = \{1, 2, \ldots, M - 1\}.$$

This reveals that $d_i(\theta, \phi) = i d_1(\theta, \phi)$, and, since $\tau_i(\theta, \phi) = \frac{d_i(\theta,\phi)}{c}$, this yields

$$\underline{H}(\omega, \theta, \phi) = \begin{bmatrix} 1 \\ e^{j\omega\frac{d_1(\theta,\phi)}{c}} \\ e^{j\omega\frac{2d_1(\theta,\phi)}{c}} \\ \vdots \\ e^{j\omega\frac{(M-1)d_1(\theta,\phi)}{c}} \end{bmatrix} = \begin{bmatrix} 1 \\ e^{j\omega\tau_1(\theta,\phi)} \\ e^{j\omega 2\tau_1(\theta,\phi)} \\ \vdots \\ e^{j\omega(M-1)\tau_1(\theta,\phi)} \end{bmatrix} = \begin{bmatrix} 1 \\ H_1(\omega, \theta, \phi) \\ H_1(\omega, \theta, \phi)^2 \\ \vdots \\ H_1(\omega, \theta, \phi)^{M-1} \end{bmatrix}. \quad (2.6)$$

Based on the coordinate system in Figure 2 and the ULA examples in Figure 3 the following $d_1(\theta, \phi)$ values can be determined,

$$\begin{aligned} \text{Endfire:} \quad & d_1(\theta, \phi) = d\cos\theta\cos\phi, \\ \text{Broadside:} \quad & d_1(\theta, \phi) = d\sin\theta, \end{aligned} \quad (2.7)$$

where d is the inter-element spacing.

2.2.4 Overall Input Signal

The source-to-array transfer functions allow for the expression of the overall array input as the superposition of the individual-source array inputs. For example, given L sources $S(\omega, \theta_l, \phi_l)$, $l = \{1, 2, \ldots, L\}$, and the corresponding source-to-array transfer functions $H(\omega, \theta_l, \phi_l)$, the array input is

$$\underline{X}(\omega) = \sum_{l=1}^{L} S(\omega, \theta_l, \phi_l) \underline{H}(\omega, \theta_l, \phi_l). \tag{2.8}$$

Since the following beamforming techniques are narrowband and operate only on a narrow band of frequencies, the following discussions suppress the explicit ω dependence.

2.3 Beamforming Methods

In order to see how $\underline{W}$ determines a beamformer's spatial filter characteristic, combine Equations 2.2 and 2.8

$$Y = \underline{W}^H \underline{X} = \underline{W}^H \left[\sum_{l=1}^{L} S(\theta_l, \phi_l) \underline{H}(\theta_l, \phi_l) \right] = \sum_{l=1}^{L} S(\theta_l, \phi_l) \underbrace{\underline{W}^H \underline{H}(\theta_l, \phi_l)}_{G(\theta_l, \phi_l)}. \tag{2.9}$$

This output expression reveals that each source $S(\theta_l, \phi_l)$ in the array input is also present in the array output, but it is scaled by the location-dependent factor known as the directional array response $G(\theta, \phi) = \underline{W}^H \underline{H}(\theta, \phi)$. This scaling factor demonstrates the effect of $\underline{W}$ upon a source from the (θ, ϕ) direction.

The overall spatial-filtering, beamforming goal is to determine a $\underline{W}$ such that $G(\theta, \phi) = 1$ for the target source and $G(\theta, \phi) = 0$ for the non-target, jammer sources. In the special case where the target and jammer source-to-array transfer functions are known [i.e., the target is at (θ_T, ϕ_T) and the jammers are at known locations $(\theta_{J,k}, \phi_{J,k})$, $k = \{1, 2, \ldots, M - 1\}$], it is possible to formulate a system of linear

equations whose solution achieves this goal:

$$
\begin{bmatrix} G(\theta_T,\phi_T) \\ G(\theta_{J,1},\phi_{J,1}) \\ \vdots \\ G(\theta_{J,M-1},\phi_{J,M-1}) \end{bmatrix} = \begin{bmatrix} \underline{H}^H(\theta_T,\phi_T) \\ \underline{H}^H(\theta_{J,1},\phi_{J,1}) \\ \vdots \\ \underline{H}^H(\theta_{J,M-1},\phi_{J,M-1}) \end{bmatrix} \underline{W} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad (2.10)
$$

where the equality $G(\theta,\phi) = \underline{W}^H\underline{H}(\theta,\phi) = \underline{H}(\theta,\phi)^H\underline{W}$ has been used.

Equation 2.10 cannot be used for general beamforming purposes, since the jammer locations are unknown and, usually, time-varying. Other methods have been developed to generate beamforming weight vectors that meet the desired goal. These techniques fall into two categories:

1. Time-Invariant Systems: in which $\underline{W}$ is fixed and optimizes a general performance criterion, such as the array directivity index.

2. Time-Varying Systems: in which $\underline{W}$ is continually updated to suit the particular beamforming environment.

Time-invariant systems have the advantages of simple implementation and robust operation, but the unchanging nature of these systems requires that the beamforming be optimized to yield generally-good performance in all potential array environments. This requirement prevents these beamformers from achieving the maximum possible level of jammer cancellation that is attainable for the particular environment in which the system is operating. Time-varying systems, on the other hand, have the advantage of continuously updating the beamforming in order to achieve the optimal performance for the particular array environment. This improved performance arises at the cost of more computationally-demanding processing, and these systems may cancel the target under certain conditions.

Since LENS processing is time-varying, this research considers only time-varying systems. The following discussion introduces these systems in three parts. First it presents the simplest concept of optimal beamforming for a given environment and introduces the robustness problems that arise with this system. It continues with a discussion of the traditional, robustness-enhancing modifications to the original, non-robust weight-vector definition. It concludes with a description of some implementations that may be used to approximate these systems.

2.3.1 Traditional, Non-Robust Beamforming

Equation 2.11 describes the traditional, non-robust frequency-domain beamforming weight-vector, which is chosen to minimize the expected output power at each frequency, $E[|Y|^2]$, subject to the constraint of preserving unit gain for the known target-source location:

$$\underline{W}_{\mathrm{NR}} = \operatorname*{argmin}_{\underline{W} \in \mathbf{C}^M} E[|Y|^2] = \operatorname*{argmin}_{\underline{W} \in \mathbf{C}^M} E[\underline{W}^H \underline{X} \underline{X}^H \underline{W}] = \operatorname*{argmin}_{\underline{W} \in \mathbf{C}^M} \underline{W}^H R_{XX} \underline{W}$$

subject to (2.11)

$$\underline{H}^H_{\text{target}} \underline{W} = 1,$$

where $R_{XX} = E[\underline{X} \underline{X}^H]$ is the input covariance matrix and $\underline{H}_{\text{target}} = \underline{H}(\theta_T, \phi_T)$ is the target-to-array transfer function vector. Solving this problem via Lagrange multipliers yields the solution:

$$\underline{W}_{\mathrm{NR}} = \frac{R_{XX}^{-1} \underline{H}_{\text{target}}}{\underline{H}^H_{\text{target}} R_{XX} \underline{H}_{\text{target}}}. \qquad (2.12)$$

In beamforming literature, this weight vector is referred to by several different names, including the Maximum Likelihood Method (MLM) weight vector and the Minumum-Variance, Distortionless Response (MVDR) weight vector. This research uses a third appelation, NR, which reflects that this is the baseline, non-robust situation of the Constrained, Minimum-Variance (CMV) weight vector [described in Section 2.3.2].

This definition of the optimal beamforming weight vector seems logical – by minimizing the expected output power while preserving the target signal, the weight vector must be attenuating the jammer sources in the environment. This system works well under ideal conditions, but, unfortunately, problems arise under more realistic conditions. These problems result from discrepancies between the assumed and actual target-to-array transfer function vectors: $\underline{H}_{target}$ and $\underline{\hat{H}}_{target}$.

The primary effect of the discrepancies between $\underline{H}_{target}$ and $\underline{\hat{H}}_{target}$ is that they can result in cancellation of the target signal – an obviously undesirable effect in any beamforming system. To see how this behavior arises, consider the case when only the target is active within the array environment. If $\underline{H}_{target}$ and $\underline{\hat{H}}_{target}$ are different, then it is possible for the beamformer to find $\underline{W}_{NR}$ such that

$$
\underline{H}^{H}_{target}\underline{W}_{NR} = 1,
$$
$$
\underline{\hat{H}}^{H}_{target}\underline{W}_{NR} = 0.
$$

In other words, the beamformer can completely eliminate the actual target signal while still preserving the assumed target-to-array transfer function.

Generalizing this result to situations in which both target and jammer sources are present, the following behavior is observed. When the target is weak relative to the jammers, i.e., the TJR (target-to-jammer power ratio) is low, then the jammer power dominates the beamforming, and the system cancels the jammer signals. When the target is strong relative to the jammers, i.e., the TJR is high, then the target dominates the beamforming and the target cancellation described above may happen.

The most obvious means of avoiding target cancellation would be to obtain a better target-to-array transfer function estimate [i.e., to reduce the discrepancies between $\underline{H}_{target}$ and $\underline{\hat{H}}_{target}$] for use in Equation 2.11. Unfortunately, this is not an easy task, given the following causes of discrepancies between $\underline{H}_{target}$ and $\underline{\hat{H}}_{target}$:

1. Array Mis-Alignment/Mis-Steer: in which the target is not located exactly at its assumed location, e.g., it might be near-field or slightly off center.

2. Element Placement Errors: in which the array elements are not located at ex-

actly the specified array locations.

3. Element Gain Mismatch: in which the array elements have unequal gains.

4. Non-Free-Field Array Operation: in which objects that are located near the array scatter the incoming wave-fronts, e.g., a hearing-aid array mounted on the head.

5. Multipath-Propagation/Reverberation: in which the target signal arrives from both the actual target location and non-target locations (wall reflections).

These effects are difficult to estimate and usually change over time, which renders it impossible to maintain a highly accurate estimate of $\underline{H}_{target}$ in practice. For this reason, modifications have been made to the definition of the optimal beamforming weight vector of Equation 2.11 in order to increase its overall robustness to target-cancellation. The following section describes one class of modifications.

2.3.2 Robust Processing: CMV Beamforming

Constrained, Minimum-Variance (CMV) beamforming is a classic approach to increasing beamforming robustness. It operates by imposing additional weight-vector constraints to the traditional, non-robust optimization of Equation 2.11:

$$\underline{W}_{CMV} = \underset{\underline{W} \in \mathbf{C}^M}{\operatorname{argmin}} \underline{W}^H R_{XX} \underline{W} \qquad (2.13)$$
$$\text{subject to:}$$
$$C^H \underline{W} = \underline{g} \quad \text{and} \quad \underline{W}^H \underline{W} \leq \delta^{-2},$$

where $(C, \underline{g})$ describes a set of linear constraints (that must include preservation of the assumed target-to-array transfer function, $\underline{H}_{target}^H \underline{W} = 1$) upon the weight vector and δ^{-2} is a quadratic constraint on the weight vector norm. Note that CMV beamforming reduces to traditional, non-robust beamforming [Equation 2.3.1] when $C = \underline{H}_{target}$, $\underline{g} = 1$, and $\delta^{-2} = \infty$. Additionally, when $\delta^{-2} = \infty$, then no quadratic

12

constraint is imposed upon the weight vector and CMV beamforming reduces to linearly-constrained, minimum-variance (LCMV) beamforming.

The solution to Equation 2.13 may be written as:

$$\underline{W}_{\mathrm{CMV}} = (R_{XX} + \epsilon I)^{-1} C [C^H (R_{XX} + \epsilon I)^{-1} C]^{-1} \underline{g}. \qquad (2.14)$$

When there is no quadratic constraint, i.e., when $\delta^{-2} = \infty$ such as for traditional, non-robust and LCMV beamforming as described above, this solution is closed-form with $\epsilon = 0$. When the quadratic constraint is active, however, this solution is not closed-form and ϵ must be adjusted iteratively until $\underline{W}^H \underline{W} \leq \delta^{-2}$.

The constraints $C^H \underline{W} = \underline{g}$ and $\underline{W}^H \underline{W} \leq \delta^{-2}$ discourage target cancellation in the following manner. The linear constraint $C^H \underline{W} = \underline{g}$ imposes restrictions on the behavior of the directional array gain $G(\theta, \phi)$. For example, the required constraint of $\underline{H}^H_{\mathrm{target}} \underline{W} = 1$ ensures that the assumed target gain is unity: $G(\theta_T, \phi_T) = 1$. Derivative constraints may be added to force $G(\theta, \phi)$ to have a flat derivative about the assumed target location, which prevents the system from canceling targets that are close to the assumed target location. Alternatively, eigenvector constraints that keep $G(\theta, \phi)$ 'near' a desired behavior about the assumed target location may be used. In general, the linear constraints are designed to increase system robustness to highly structured deviations (e.g., array misalignment) between the actual and assumed target-to-array transfer function.

The norm constraint, $\underline{W}^H \underline{W} \leq \delta^{-2}$, imposes restrictions on the white-noise-gain of the system. Specifically, the white noise gain, G_w measures the improvement in the output relative to the input TJR (target-to-jammer power ratio) when no directional jammers are present and only uncorrelated noise is introduced at each array element:

$$G_w = \frac{|\underline{H}^H_{\mathrm{target}} \underline{W}|^2}{\underline{W}^H \underline{W}} = \frac{1}{\underline{W}^H \underline{W}}, \qquad (2.15)$$

where the fact that $\underline{H}^H_{\mathrm{target}} \underline{W} = 1$ has been used. By maintaining $\underline{W}^H \underline{W} \leq \delta^{-2}$, then $G_w \geq \delta^2$, which guarantees a minimum level of beamforming improvement, even in the presence of uncorrelated sensor noise. This norm constraint increases system

robustness to factors such as array-element placement error, which are independent and may be regarded as additive uncorrelated noise at the array input.

2.3.3 Traditional Implementations

In general, the array environment and input covariance matrix are unknown and time-varying, which prevents direct solution for the optimal CMV weight vector as described in Equation 2.13. Approaches have been designed, however, that approximate the optimal solution and that track the changing array environment. This section introduces three such implementations. The first two implementations are possible only when $\delta^{-2} = \infty$, such as for traditional non-robust and LCMV beamforming sub-classes of CMV beamforming described above. The third technique is more general and may be used with any CMV formulation.

All implementations use a temporal stream of array-input observations, $\{\underline{X}^{(l)}\}$, to derive a corresponding temporal stream of beamforming weight vectors, $\{\underline{W}^{(l)}_{CMV}\}$. For the broadband processing simulated in much of this research, the narrowband $\underline{X}^{(l)}$ arise through block FFT processing on the broadband input.

Direct Matrix Inversion: In the absence of a weight vector norm constraint, i.e., when $\delta^{-2} = \infty$, the solution of Equation 2.14 is closed form. In this case, $\underline{W}^{(l)}_{CMV}$ may be obtained by forming an input-covariance estimate $\hat{R}^{(l)}_{XX}$

and using this estimate in Equation 2.14. This method is known as the Direct Matrix Inversion or DMI method, because it involves the inversion of the estimated covariance matrix. The main advantage of this implementation is its relatively fast adaptation time to a changing array environment: adaptation time is governed entirely by the delays associated with the covariance matrix estimation process and are independent of the array environment. Unfortunately, this fast adaptation comes at the cost of matrix inversions, which can be computationally expensive [$O(M^3)$ operations].

14

Frost Processing: An alternative implementation to the DMI implementation, also for the special case when $\delta^{-2} = \infty$, is a Least-Mean-Squared (LMS) based adaptive approach that uses a gradient search to converge upon the optimal beamforming weight vector. Specifically, the weight vector follows the gradient towards minimal output power, but it is projected back into the constraint subspace (defined by $C^H \underline{W} = \underline{g}$) at each iteration:

$$\underline{W}_{CMV}^{(l+1)} = \underline{W}_c + \tilde{P}_c \left[I - \mu \underline{X}^{(l)} \underline{X}^{(l)H} \right] \underline{W}_{CMV}^{(l)}, \tag{2.16}$$

where

$$\underline{W}_c = C(C^H C)^{-1} \underline{g} \quad \text{and} \quad \tilde{P}_c = I - C(C^H C)^{-1} C^H.$$

This processing has the advantage of being less complex [$O(M^2)$ operations] to implement than DMI, but this complexity advantage comes at the cost of longer adaptation times: adaptation time is proportional to $\frac{1}{\lambda_{min}}$, where λ_{min} is the smallest eigenvalue of the underlying input R_{XX}, which means that certain array environments can exhibit very long adaptation times. Details of this technique are set forth in the paper entitled "An Algorithm for Linearly Constrained Adaptive Array Processing", Proc. IEEE, 60:926-935, August, 1972, O.L. Frost.

Scaled Projection Algorithm: The processing set forth above may be altered yield an adaptive beamformer implementation that includes the weight vector norm constraint, $\underline{W}^H \underline{W} \leq \delta^{-2}$. In this case, the basic Frost algorithm is used, but the weight vector is restricted to obey $\underline{W}^H \underline{W} \leq \delta^{-2}$ at each step:

Step 1: $\tilde{\underline{V}}^{(l+1)} = \tilde{P}_c \left[\underline{V}^{(l)} - \mu \underline{X}^{(l)} \underline{X}^{(l)H} \underline{W}_{CMV}^{(l)} \right]$

Step 2: $\underline{V}^{(l+1)} = \begin{cases} \tilde{\underline{V}}^{(l+1)}, & \tilde{\underline{V}}^{(l+1)H} \tilde{\underline{V}}^{(l+1)} \leq b^2 \\ \frac{b\tilde{\underline{V}}^{(l+1)}}{|\tilde{\underline{V}}^{(l+1)}|}, & \tilde{\underline{V}}^{(l+1)H} \tilde{\underline{V}}^{(l+1)} > b^2 \end{cases}$

Step 3: $\underline{W}_{CMV}^{(l+1)} = \underline{W}_c + \underline{V}_{l+1}$ $\qquad$ (2.17)

where: $\underline{W}_c = C(C^H C)^{-1} \underline{g}, \quad \tilde{P}_c = I - C(C^H C)^{-1} C,$

and $b^2 = \delta^{-2} - \underline{g}^H (C^H C)^{-1} \underline{g}.$

As with Frost processing, this processing has the advantage of being relatively simple to implement [$O(M^2)$ operations] but it may exhibit long adaptation times.

As set forth above, a significant about of care (i.e., constraints) must be taken when defining the optimal beamforming weight vector in order to yield a solution that is robust to situations that may cause target cancellation. The CMV definition provides one possible solution to this problem, but it suffers from some disadvantages.

1. The CMV constraints are indirect – their impact upon the beamforming performance and their prevention of target-cancellation can be difficult to understand. This renders the overall constraint-selection process confusing.

2. Each linear CMV constraint in (C, g) reduces the beamformer degrees of freedom by one. Under good beamforming conditions, e.g., when the TJR is low, this reduction in the degrees of freedom limits the potential jammer cancellation.

Therefore, there is a need for a robustness-control mechanism, which yields a system that has direct and flexible control over the beamforming process.

SUMMARY OF THE INVENTION

Briefly, according to the present invention, an adaptive multi-tap frequency domain digital filter processes an input signal vector X from a plurality of spatially separated transducers that detect energy from a plurality of sources including a target energy source and at least one non-target energy source. The filter receives and processes the input signal vector X to attenuate noise from non-target sources and provides an output signal vector Y. Tap weights W_N for filter are selected by first parameterizing each of the tap weights W_N such that each of the tap weights W_N is characterized by a vector of parameters $\underline{\beta}_{opt}$, and the solving for each parameter of the vector $\underline{\beta}_{opt}$ by minimizing the expected power of the array output signal Y. A robustness-control transformation is then applied to the vector $\underline{\beta}_{opt}$ to provide a robust vector $\underline{\beta}_{rob}$, wherein the robustness-control transformation identifies and reduces target canceling components of the vector $\underline{\beta}_{opt}$ while preserving non-target canceling components. Finally, the weight vector indicative of the filter tap weights is formed as a function of the vector $\underline{\beta}_{rob}$.

The tap weight selection technique of the present invention preferably executes in real-time. The technique may be used with transducer arrays such as microphones (e.g., hearing aides, sonar processing, etc....) and radar receivers where it is desirable to attenuate signal contributions associated with non-target signal sources.

Beam forming parameterization identifies parameters of the vector $\underline{\beta}_{opt}$ that describe the nulling behavior of the tap weights W_N.

The present invention uses a direct and flexible robustness control mechanism to yield a beamformer that provides good performance and it robust to a wide variety of adverse conditions. Notably, the present invention includes a novel robustness-control mechanism, which parameterizes the beamforming problem and yields a system that has direct and more flexible control over the beamforming process.

Advantageously, selecting the tap weight of the adaptive filter according to the present invention nulls (i.e., attenuates) signal components associated with non-target noise sources, while

17

retaining signal components associated with a desired target source(s).

These and other objects, features and advantages of the present invention will become more apparent in light of the following detailed description of preferred embodiments thereof, as illustrated in the accompanying drawings.

5

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram illustration of a beamforming array;

Figure 2 is a plot of a coordinate system;

Figure 3A and 3B (collectively Figure 3) are simplified pictorial illustrations of two uniform linear array (ULA) configurations with respect to a target;

Figure 4 is a block diagram interpretation of the processing according to the present invention for a four element ULA;

Figure 5 is a plot of the transformation of $\beta_{opt,j}$ to $\beta_{rob,j}$;

Figure 6 is a pictorial illustration of the distance metric Ω between the vector $\underline{Z}$ and the subspace S;

Figure 7 is a plot of the contours of various values of $\Omega_i(\beta)$ for ULA arrays having 2, 3, 4 and 5 elements;

Figure 8 is a plot that illustrates three LENS parameter classification regions for an array having three elements as described by Equation 3.20 with $\Omega_r = 30°$ and $\Omega_r = 60°$;

Figure 9 is a plot of the contours for a three element array and $\Omega(\beta_i) = 30°, 40°, 50°$ and $60°$ with β_i indicated as (x);

Figure 10 is a plot that illustrates three LENS parameter classification regions for a three element array according to Equation 3.23 when R_r and R_j are chosen to reflect the $R_T = 30°$ and $R_J = 60°$ contours, respectively;

Figure 11 is a plot of illustrating four examples of $\beta_{opt,j}$ for a two-element array along with the corresponding LENS robustness regions;

Figure 12 is a flow chart illustration of processing steps according to the present invention;

Figure 13 is a flow chart illustration of alternative embodiment processing steps according to the present invention;

Figure 14 is a flow chart illustration of yet another series of alternative embodiment processing steps according to the present invention;

Figure 15 is a block diagram illustration of processing steps for solving for $\underline{\beta}_{opt}$ given a known covariance matrix;

Figures 16A and 16b (collectively Figure 16) are block diagram illustrations of processing steps regarding the adaptive relaxation based solutions; and

5   Figure 17 is a plot that illustrates the relative computational complexity for the complete update relaxation, partial update relaxation and gradient LENS implementations.

10

DETAILED DESCRIPTION OF THE INVENTION

3   LENS Beamforming Processing

5   The present invention is directed to a beamforming technique that modifies traditional non-robust beamforming to increase beamforming robustness by preventing target cancellation. Significantly, unlike the prior art, the present invention solves for the robust weight vector indirectly by dividing the beamforming process into three components/steps.

:0   Like CMV beamforming [Section 2.3.2], Location-Estimating, Null-Steering (LENS) beamforming modifies traditional, non-robust beamforming [Section 2.3.1] in an attempt to increase beamforming robustness by preventing target cancellation. LENS differs from CMV processing however, in that it does not solve for a robust beamforming weight vector directly. Instead, it solves for the robust weight vector indirectly

15   by dividing the beamforming process into three components:

1. Estimation of the optimal LENS parameter set, $\underline{\beta}_{opt}$, that describes the traditional, non-robust beamforming problem.

2. Transformation of $\underline{\beta}_{opt}$ into a 'robust' parameter set, $\underline{\beta}_{rob}$, that is designed to ensure robust beamforming and to prevent target cancellation.

20   3. Use of $\underline{\beta}_{rob}$ to generate a final beamforming weight vector.

It should be noted that LENS processing assumes that $\underline{H}_{target} = [1, 1, \ldots, 1]^T = \underline{1}$. This target-to-array transfer functions arises naturally for broadside ULAs [Section 2.2.2]. For more general arrays, it may be realized by scaling each array input i by the steering factor $\frac{1}{\{\underline{H}_{target}\}_i}$, where $\{\underline{H}_{target}\}_i$ is the ith element of $\underline{H}_{target}$.

25   The primary advantage of the LENS parameterization is that it allows for the use of an innovative and direct method of beamforming control. Consider as an example Uniform Linear Arrays (ULAs), for which each parameter $\beta_{opt,i}$ in $\underline{\beta}_{opt}$ determines a

beamforming null location in the array environment [Section 3.3.1]. In the process of transforming $\underline{\beta}_{opt}$ into $\underline{\beta}_{rob}$, it is possible to control where the final beamforming weight vector will place its nulls [Section 3.3.2] and to prevent target cancellation by restricting these nulls to be a certain angular distance away from the assumed target location. (It is this manifestation of the algorithm that leads to the Location-Estimating, Null-Steering appellation.)

This chapter develops the basic components of LENS beamforming. It begins by introducing the the LENS weight vector and parameter set. It then determines the optimum LENS parameter set to describe traditional, non-robust beamforming and defines the LENS robustness transformation. It concludes by drawing these components together into the overall, general LENS algorithm.

3.1   The LENS Beamforming Weight Vector

The key component of LENS processing is the LENS beamforming weight vector, $\underline{W}_{\text{LENS},M}(\underline{\beta})$. For an M-element array, this weight vector is determined from the $M - 1$ LENS parameters, $\underline{\beta} = [\beta_1, \beta_2, \ldots, \beta_{M-1}]$, as follows:

$$\underline{W}_{\text{LENS},M}(\underline{\beta})^H = \left(\frac{1}{2}\right)^{M-1} [\ (1 - j\beta_1)\ \ (1 + j\beta_1)\ ] * \tag{3.1}$$

$$[\ (1 - j\beta_2)\ \ (1 + j\beta_2)\ ] * \cdots * [\ (1 - j\beta_{M-1})\ \ (1 + j\beta_{M-1})\ ],$$

where the '$*$' operation denotes row vector convolution. It turns out that the LENS parameters $\underline{\beta}$ provide a convenient description of the beamforming that is realized by

$\underline{W}_{\text{LENS},M}(\underline{\beta})$ and, consequently, they may be used to control the beamforming and to prevent target cancellation.

In order to understand how $\underline{W}_{\text{LENS},M}(\underline{\beta})$ arises and to obtain a 'first look' at the relationship between $\underline{\beta}$ and $\underline{W}_{\text{LENS},M}(\underline{\beta})$, consider the two-element $\underline{W}_{\text{LENS},2}(\beta_1)$:

$$\underline{W}_{\text{LENS},2}(\beta_1)^H = \left[\begin{array}{cc} \frac{1-j\beta_1}{2} & \frac{1+j\beta_1}{2} \end{array} \right]. \tag{3.2}$$

Defining ψ such that

$$\psi = \frac{\beta_1 + j}{\beta_1 - j} \quad \Longleftrightarrow \quad \beta_1 = j\frac{\psi+1}{\psi-1}, \tag{3.3}$$

where both ψ and β_1 may be complex, and substituting into $\underline{W}_{\text{LENS},2}(\beta_1)$ yields

$$\underline{W}_{\text{LENS},2}(\beta_1)^H = \left[\begin{array}{cc} \frac{-\psi}{1-\psi} & \frac{1}{1-\psi} \end{array} \right],$$

where $\underline{W}_{\text{LENS},2}(\beta_1)^H \underline{H}_{\text{target}} = 1$ and $\underline{W}_{\text{LENS},2}(\beta_1)^H \underline{H}_{\psi} = 0$, with

$$\underline{H}_{\psi} = \left[\begin{array}{c} 1 \\ \psi \end{array} \right].$$

The one-to-one relationship between β_1 and ψ shows that β_1 completely describes the nulling behavior of $\underline{W}_{\text{LENS},M}(\underline{\beta})$.

The precise nulling behavior depends upon the actual source-to-array transfer functions $\underline{H}(\theta, \phi)$ for the array in question. Consider the special case of far-field sources, for which the source-to-array transfer functions are given by Equation 2.6:

$$\underline{H}(\theta, \phi) = \left[\begin{array}{c} 1 \\ e^{j\omega\tau(\theta,\phi)} \end{array} \right].$$

By equating $\psi = e^{j\omega\tau(\theta,\phi)}$ and substituting into Equation 3.3, then $\underline{W}_{\text{LENS},M}(\underline{\beta})$

clearly nulls source locations (θ, ϕ) that satisfy:

$$\beta_1 = j \frac{e^{j\omega\tau(\theta,\phi)}+1}{e^{j\omega\tau(\theta,\phi)}-1} = j \frac{e^{j\frac{\omega\tau(\theta,\phi)}{2}}+e^{-j\frac{\omega\tau(\theta,\phi)}{2}}}{e^{j\frac{\omega\tau(\theta,\phi)}{2}}-e^{-j\frac{\omega\tau(\theta,\phi)}{2}}} = \cot\left(\frac{\omega\tau(\theta,\phi)}{2}\right). \qquad (3.4)$$

In this special case, since $\tau(\theta, \phi)$ is real, β_1 is also real. As discussed in Section 3.3.1, for general M-element ULA architectures with far-field sources, each LENS parameter β_i determines one beamforming null location through a similar relation to Equation 3.4. For more general non-ULA array architectures and for near-field sources, the relationship is more complex.

The iterated-convolution, M-element LENS weight vector, $\underline{W}_{\text{LENS},M}(\underline{\beta})$, may be interpreted as a cascade of the two-element LENS weight vectors [Equation 3.2]. Figure 4 shows this cascade for the special case of a ULA array architecture with far-field sources, for which each LENS parameter and, consequently, each stage in the cascade produces one null in the array directional response [Section 3.3.1].

3.1.1 LENS Weight Vector: a Re-Formulation

The LENS beamforming weight vector $\underline{W}_{\text{LENS},M}(\underline{\beta})$ may be re-expressed in a manner that simplifies the analysis of the LENS algorithm. It can be shown that $\underline{W}_{\text{LENS},M}(\underline{\beta})$ is equal to the linear transformation T_M of another vector $\underline{U}_{\text{LENS},M}(\underline{\beta})$:

$$\underline{W}_{\text{LENS},M}(\underline{\beta})^H = \underline{U}_{\text{LENS},M}(\underline{\beta})^H T_M^H, \qquad (3.5)$$

where $\underline{U}_{\text{LENS},M}(\underline{\beta})$ is defined as

$$\underline{U}_{\text{LENS},M}(\underline{\beta})^H = \left(\frac{1}{2}\right)^{M-1} [\ 1 \quad \beta_1\ ] * [\ 1 \quad \beta_2\ ] * \cdots * [\ 1 \quad \beta_{M-1}\ ], \qquad (3.6)$$

and T_M is given by

$$T_M^H = \begin{bmatrix} [\,1\ \ 1\,] & * & [\,1\ \ 1\,] & * & \cdots & * & [\,1\ \ 1\,] & * & [\,1\ \ 1\,] \\ [\,1\ \ 1\,] & * & [\,1\ \ 1\,] & * & \cdots & * & [\,1\ \ 1\,] & * & [\,-j\ \ j\,] \\ & & \vdots & & & & \vdots & & \vdots \\ [\,1\ \ 1\,] & * & [\,-j\ \ j\,] & * & \cdots & * & [\,-j\ \ j\,] & * & [\,-j\ \ j\,] \\ [\,-j\ \ j\,] & * & [\,-j\ \ j\,] & * & \cdots & * & [\,-j\ \ j\,] & * & [\,-j\ \ j\,] \end{bmatrix}. \qquad (3.7)$$

$$\underbrace{\hspace{8cm}}_{k-1 \text{ pairs per row}}$$

3.2 Optimal Beamforming Re-Expressed as LENS

The LENS weight vector allows for the LENS parameterization of traditional, non-robust beamforming [Section 2.3.1]. Specifically, define $\underline{\beta}_{opt}$ to minimize the expected power in the LENS array output signal $Y = \underline{W}_{\text{LENS},M}(\underline{\beta})^H \underline{X}$:

$$\underline{\beta}_{opt} = \underset{\underline{\beta} \in \mathbf{C}^{M-1}}{\operatorname{argmin}} E[|Y(\underline{\beta})|^2] = \underset{\underline{\beta} \in \mathbf{C}^{M-1}}{\operatorname{argmin}} \underline{W}_{\text{LENS},M}(\underline{\beta})^H R_{XX} \underline{W}_{\text{LENS},M}(\underline{\beta}). \qquad (3.8)$$

This $\underline{\beta}_{opt}$ yields a $\underline{W}_{\text{LENS},M}(\underline{\beta}_{opt})$ that is equal to the traditional, non-robust $\underline{W}_{\text{NR}}$ from Equation 2.11. To see why this is so, consider the following proof.

Given that $\underline{\beta}_{opt}$ satisfies Equation 3.8, then $\underline{W}_{\text{LENS},M}(\underline{\beta}_{opt}) = \underline{W}_{\text{NR}}$, where $\underline{W}_{\text{NR}}$ satisfies Equation 2.11 in the case where steering delays have been applied to the array so that $\underline{H}_{\text{target}} = \underline{1}$:

$$\underline{W}_{\text{NR}} = \underset{\underline{W} \in \mathbf{C}^{M-}}{\operatorname{argmin}} \underline{W}^H R_{XX} \underline{W} \qquad (3.9)$$
$$\text{subject to:} \ \underline{1}^H \underline{W} = 1.$$

Define the set $\mathcal{A}$ of M-dimensional beamforming weight vectors that are represented by $\underline{W}_{\text{LENS},M}(\underline{\beta})$ as $\underline{\beta}$ spans the space of $M-1$-dimensional complex vectors, $\mathbf{C}^{M-1}$:

$$\mathcal{A} = \{\underline{W}_{\text{LENS},M}(\beta) \in \mathbf{C}^M | \underline{\beta} \in \mathbf{C}^{M-1}\}.$$

It can be shown that $\mathcal{A}$ is equivalent to the set $\mathcal{B}$ of all M-dimensional complex vectors $\underline{W}$ that sum to 1:

$$\mathcal{B} = \{\underline{W} \in \mathbf{C}^M | \underline{1}^H \underline{W} = 1\},$$

where $\underline{1} = [1, 1, \ldots, 1]^T$. Hence $\mathcal{A} = \mathcal{B}$.

Using the set $\mathcal{B}$, Equation 3.9 may be re-written as:

$$\underline{W}_{\text{NR}} = \operatorname*{argmin}_{\underline{W} \in \mathbf{C}^M} \underline{W}^H R_{XX} \underline{W},$$

where $\underline{W}$ clearly varies over the entire set $\mathcal{B}$ for the optimization. Repeating Equation 3.8, the optimal LENS parameter set is:

$$\underline{\beta}_{opt} = \operatorname*{argmin}_{\underline{\beta} \in \mathbf{C}^{M-1}} \underline{W}_{\text{LENS},M}(\underline{\beta})^H R_{XX} \underline{W}_{\text{LENS},M}(\underline{\beta}),$$

where $\underline{W}_{\text{LENS},M}(\underline{\beta})$ varies over the entire set $\mathcal{A}$ for the optimization.

Since (1) $\underline{W}$ varies over $\mathcal{B}$, (2) $\underline{W}_{\text{LENS},M}(\underline{\beta})$ varies over $\mathcal{A}$, and (3) $\mathcal{A} = \mathcal{B}$, then

$$\underline{W}_{\text{LENS},M}(\underline{\beta}_{opt}) = \underline{W}_{\text{NR}}.$$

3.2.1 Closed-Form LENS Solution

The equality of $\underline{W}_{\text{LENS},M}(\underline{\beta})$ and $\underline{W}_{\text{NR}}$, as described above, and the LENS reformulation from Section 3.1.1 may be combined to yield a closed-form expression for $\underline{\beta}_{opt}$. Specifically, using the closed-form solution for $\underline{W}_{\text{NR}}$ [Equation 2.12] and

recalling that $\underline{H}_{target} = \underline{1}$, then

$$\underline{W}_{LENS,M}(\underline{\beta}_{opt})^H = \underline{W}_{NR}^H = \left(\frac{R_{XX}^{-1}\underline{1}}{\underline{1}^H R_{XX}^{-1}\underline{1}}\right)^H = \frac{\underline{1}^H R_{XX}^{-1}}{\underline{1}^H R_{XX}^{-1}\underline{1}}.$$

Now, recall that $\underline{W}_{LENS,M}(\underline{\beta}_{opt})^H = \underline{U}_{LENS,M}(\underline{\beta}_{opt})^H T_M^H$ [Equation 3.5] and it is known that $Q_M = T_M^{-1}$ exists, which leads to

$$\underline{U}_{LENS,M}(\underline{\beta}_{opt})^H = \underline{W}_{LENS,M}(\underline{\beta}_{opt})^H Q_M^H = \frac{\underline{1}^H R_{XX}^{-1} Q_M^H}{\underline{1}^H R_{XX}^{-1}\underline{1}}.$$

Finally, use the fact that the elements of the vector $\underline{U}_{LENS,M}(\underline{\beta})^H$ form the coefficients of a polynomial that has roots at $-\beta_i$ [Appendix B] to obtain the expression for $\underline{\beta}_{opt}$:

$$\underline{\beta}_{opt} = -\text{roots}\left(\underline{U}_{LENS,M}(\underline{\beta}_{opt})^H\right) = -\text{roots}\left(\frac{\underline{1}^H R_{XX}^{-1} Q_M^H}{\underline{1}^H R_{XX}^{-1}\underline{1}}\right). \qquad (3.10)$$

Equation 3.10 reveals that $\underline{\beta}_{opt}$ may be obtained by inverting R_{XX} and performing a polynomial root solution on the polynomial described by $\underline{U}_{LENS M}(\underline{\beta})$.

3.3 Transforming $\underline{\beta}_{opt}$ into $\underline{\beta}_{rob}$

As shown above, $\underline{\beta}_{opt}$ is a parameterization of the traditional, non-robust beam-forming of Section 2.3.1. This means that a beamformer that uses $\underline{W}_{LENS,M}(\underline{\beta}_{opt})$ would exhibit all of the non-robust, target-canceling characteristics of the original, non-robust approach. The key feature that differentiates LENS beamforming from traditional, non-robust beamforming is the robustness-control transformation of $\underline{\beta}_{opt}$ into $\underline{\beta}_{rob}$. This transformation uses the relationship between $\underline{\beta}$ and the nulling behavior of $\underline{W}_{LENS,M}(\underline{\beta})$ to identify and eliminate any target-canceling components of $\underline{\beta}_{opt}$ while preserving the jammer-canceling components.

The robustness transformation is achieved by classifying each optimal LENS pa-

27

rameter $\beta_{opt,i}$ into one of three sets and then transforming it according to the rule:

$$\beta_{rob,i} = \begin{cases} 0 & \beta_{opt,i} \in \mathcal{T} \\ \beta_{opt,i} & \beta_{opt,i} \in \mathcal{J} \\ F[\beta_{opt,i}] & \beta_{opt,i} \in \overline{\mathcal{T} \cup \mathcal{J}} \end{cases} \qquad (3.11)$$

where $\mathcal{T}$ and $\mathcal{J}$ are the 'target-canceling' and 'jammer-canceling' LENS-parameter regions, respectively. Figure 5 demonstrates this process graphically for parameters in each of the three classes. For $\beta_{opt,1} \in \mathcal{T}$, the LENS parameter contains primarily target-canceling information, and so $\beta_{rob,i}$ is set to 0, which is a 'safe' non-target-canceling parameter value. For $\beta_{opt,2} \in \mathcal{J}$, the LENS parameter contains primarily jammer-canceling information, and so $\beta_{rob,i}$ is set to the unaltered $\beta_{opt,i}$. For $\beta_{opt,3} \in \overline{\mathcal{T} \cup \mathcal{J}}$, the LENS parameter is in an intermediate region and $\beta_{rob,i}$ is set to $F[\beta_{opt,i}]$, which is a smooth transition between the $\beta_{opt,i} \in \mathcal{T}$ and $\beta_{opt,i} \in \mathcal{J}$ extremes.

Equation 3.11 reveals that the main element in the $\underline{\beta}_{opt} \to \underline{\beta}_{rob}$ transformation is the parameter classification $\mathcal{T}$, $\mathcal{J}$, and $\overline{\mathcal{T} \cup \mathcal{J}}$. The following discussion develops this classification in two stages. The first stage explores LENS parameters and investigates the beamforming information contained in $\underline{\beta}$, while the second stage uses this information to derive the actual parameter classes. These two components are then combined into the final LENS robustness transformation, and some simple example situations are explored to demonstrate its operation.

3.3.1 LENS Parameter Classification, Part 1: Beamforming significance of $\underline{\beta}$

In order to classify a LENS parameter as either target-canceling or jammer-canceling, it is necessary to understand the significance of $\underline{\beta}$ as it relates to the overall system beamforming behavior. This relationship is stated as a fact for the two-element array of Section 3.1 [Equation 3.3], but it is more complex for general, M-element array architectures.

For general array architectures, $\underline{\beta}$ and $\underline{W}_{\text{LENS},M}(\underline{\beta})$ are related by the fact that $\underline{\beta}$ describes a basis for the null-space, $\mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta})]$, of $\underline{W}_{\text{LENS},M}(\underline{\beta})$, which in turn describes the source-to-array transfer functions that are nulled by $\underline{W}_{\text{LENS},M}(\underline{\beta})$. Given $\underline{\beta} = [\beta_1, \beta_2, \ldots, \beta_{M-1}]^T$, this basis consists of the vectors

$$\underline{V}_i(\underline{\beta}) = \begin{bmatrix} 1 \\ \left(\frac{\beta_i+j}{\beta_i-j}\right) \\ \left(\frac{\beta_i+j}{\beta_i-j}\right)^2 \\ \vdots \\ \left(\frac{\beta_i+j}{\beta_i-j}\right)^{M-1} \end{bmatrix}, \quad i = \{1, 2, \ldots, M-1\}. \tag{3.12}$$

To see why $\{\underline{V}_i(\underline{\beta})\}$ forms a basis for $\mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta})]$, consider the following:

1. $\mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta})]$ is an $(M-1)$-dimensional space: This arises from the fact that the null-space of a single M-element vector is always $(M-1)$-dimensional.

29

2. span$[\underline{V}_i(\beta)]$ is an $(M-1)$-dimensional space: Use the $M-1$ vectors $\underline{V}_i(\beta)$ and the vector H_{target} to form the columns of an M-by-M Vandermonde matrix. Assuming that the β_i are unique , then this Vandermonde matrix is non-singular, and its columns, including the $\underline{V}_i(\beta)$, are linearly-independent . These $M-1$, linearly-independent $\underline{V}_i(\beta)$ then span an $M-1$-dimensional space.

3. span$[\underline{V}_i(\beta)] \subseteq \mathcal{N}[\underline{W}_{\mathrm{LENS},M}(\underline{\beta})]$:   It can be shown that each $\underline{V}_i(\beta)$ is an element of $\mathcal{N}[\underline{W}_{\mathrm{LENS},M}(\underline{\beta})]$, i.e., that $\underline{W}_{\mathrm{LENS},M}(\underline{\beta})^H \underline{V}_i(\underline{\beta}) = 0$. For this reason, any vector that may be expressed as the weighted sum of $\underline{V}_i(\beta)$ – that is to say any vector within span$[\underline{V}_i(\beta)]$ – is also within $\mathcal{N}[\underline{W}_{\mathrm{LENS},M}(\underline{\beta})]$.

The only way that one $(M-1)$-dimensional space may contain another $(M-1)$-dimensional space is if the two spaces are equal, and so these three properties reveal that span$[\underline{V}_i(\beta)] = \mathcal{N}[\underline{W}_{\mathrm{LENS},M}(\underline{\beta})]$. This is equivalent to stating that the vectors $\underline{V}_i(\beta)$ provide a basis for $\mathcal{N}[\underline{W}_{\mathrm{LENS},M}(\underline{\beta})]$.

This leads to the conclusion that the beamforming weight vector $\underline{W}_{\mathrm{LENS},M}(\underline{\beta})$ will null any source-to-array transfer function $\underline{H}(\theta,\phi)$ for which there exists a set of complex coefficients, $\{c_i\}$, such that:

$$\underline{H}(\theta,\phi) = \sum_{i=1}^{M-1} c_i \underline{V}_i(\underline{\beta}). \tag{3.13}$$

Since the target-to-array transfer function $\underline{H}_{target} = \underline{1}$ is known, it is possible to use the $\underline{V}_i(\beta)$ to determine how 'close' $\underline{H}_{target}$ is to $\mathcal{N}[\underline{W}_{\mathrm{LENS},M}(\underline{\beta})] = $ span$[\underline{V}_i(\beta)]$ and to define the desired LENS-classification regions $\mathcal{T}$, $\mathcal{J}$, and $\overline{\mathcal{T} \cup \mathcal{J}}$. Before proceeding

30

with this process, however, it is interesting to consider the special case of ULA beamformer architectures with far-field sources, which display a convenient correspondence between the LENS parameters and the beamformer null locations.

Special Case: ULA Architectures

Recall the ULA source-to-array transfer functions for far-field sources

$$\underline{H}_{ULA}(\theta,\phi) = \begin{bmatrix} 1 \\ e^{j\omega\tau(\theta,\phi)} \\ \left(e^{j\omega\tau(\theta,\phi)}\right)^2 \\ \vdots \\ \left(e^{j\omega\tau(\theta,\phi)}\right)^{M-1} \end{bmatrix} = \begin{bmatrix} 1 \\ \underline{H}_1(\theta,\phi) \\ \underline{H}_1(\theta,\phi)^2 \\ \vdots \\ \underline{H}_1(\theta,\phi)^{M-1} \end{bmatrix}, \qquad (3.14)$$

where $\tau(\theta,\phi)$ is the additional temporal delay experienced by a source from (θ,ϕ) when traveling from array element i to array element $(i+1)$. It turns out that, for this special type of source-to-array transfer function, $\underline{W}_{LENS,M}(\underline{\beta})$ nulls only source locations with $\underline{H}(\theta,\phi)$ exactly equal to one of the basis vectors $\underline{V}_i(\underline{\beta})$:

$$\underline{H}_{ULA}(\theta,\phi) \in \mathcal{N}[\underline{W}_{LENS,M}(\underline{\beta})] \iff \underline{H}_{ULA}(\theta,\phi) = \underline{V}_i(\underline{\beta})$$
$$\text{for some } i \in \{1,2,\ldots,M-1\}. \qquad (3.15)$$

The $\Longleftarrow$ property of this relation is straightforward to see: since $\underline{V}_i(\underline{\beta})$ is a basis vector for $\mathcal{N}[\underline{W}_{LENS,M}(\underline{\beta})]$ then $\underline{H}_{ULA}(\theta,\phi) = \underline{V}_i(\underline{\beta})$ is clearly within $\mathcal{N}[\underline{W}_{LENS,M}(\underline{\beta})]$. The $\Longrightarrow$ property is less obvious, however. Consider the M-by-M A matrix formed using the $M-1$ vectors $\underline{V}_i$ and the vector $\underline{H}_{ULA}(\theta,\phi) \in \mathcal{N}[\underline{W}_{LENS,M}(\underline{\beta})]$:

$$A = \begin{bmatrix} \underline{V}_1(\underline{\beta}) & | & \cdots & | & \underline{V}_{M-1}(\underline{\beta}) & | & \underline{H}_{ULA}(\theta,\phi) \end{bmatrix}.$$

Since $\{\underline{V}_i(\underline{\beta})\}$ forms a basis for $\mathcal{N}[\underline{W}_{LENS,M}(\underline{\beta})]$, then $\underline{H}_{ULA}(\theta,\phi) \in \mathcal{N}[\underline{W}_{LENS,M}(\underline{\beta})]$ must be linearly dependent upon the $\underline{V}_i(\underline{\beta})$, which means that A is singular. Given the structure of $\underline{V}_i(\underline{\beta})$ [Equation 3.12] and $\underline{H}_{ULA}(\theta,\phi)$ [Equation 3.14]), then A is a

Vandermonde matrix, which is singular only if two of its columns are equal. Since the β_i and $\underline{V}_i(\underline{\beta})$ are unique, then the Vandermonde A is singular only if $\underline{H}_{ULA}(\theta, \phi) = \underline{V}_i(\underline{\beta})$ for some i, which proves $\Longrightarrow$.

5   By equating the elements of $\underline{V}_i(\underline{\beta})$ and $\underline{H}(\theta, \phi)$, Equation 3.15 yields

$$H_1(\theta, \phi) = e^{j\omega\tau(\theta,\phi)} = \frac{\beta_i + j}{\beta_i - j}, \quad \text{for some } i = 1, 2, \ldots, M - 1, \qquad (3.16)$$

which may be re-written as:

$$\beta_i = \cot\left(\frac{\omega\tau(\theta,\phi)}{2}\right), \quad \text{for some } i = 1, 2, \ldots, M - 1. \qquad (3.17)$$

Solving this relationship for each β_i reveals that $\underline{W}_{LENS,M}(\underline{\beta})$ nulls sources with 10   locations that satisfy:

$$(\theta_i, \phi_i) = \tau^{-1}\left(\frac{2}{\omega}\cot^{-1}\beta_i\right), \quad i = 1, 2, \ldots, M - 1. \qquad (3.18)$$

Note that these results match those stated in Equation 3.4 for two-element arrays

Given that $\underline{\beta}_{opt}$ minimizes the expected beamformer output [Equation 3.8], then 15 $\underline{W}_{LENS,M}(\underline{\beta}_{opt})$ should null the non-target sources in the array environment. This knowledge together with Equations 3.17 and 3.18 suggest that the $\underline{\beta}_{opt,i}$ are, in some sense, a set of non-target location estimates.

It should be noted that the special ULA relationship between the physical null location for $\underline{W}_{LENS,M}(\underline{\beta})$ and the LENS parameter set $\underline{\beta}$ holds for any array architecture that has 20   a source-to-array transfer function of the form:

$$\underline{H}(\theta, \phi) = \begin{bmatrix} 1 \\ H_1(\theta, \phi) \\ H_1(\theta, \phi)^2 \\ \vdots \\ H_1(\theta, \phi)^{M-1} \end{bmatrix}.$$

Generally, only free-field ULAs will exhibit this form of source-to-array transfer function for all source locations. For other arrays, however, it is possible to pre-process the array input to yield source-to-array transfer function structures that are similar to the ULA structure for a subset of source locations that are of interest for the particular beamforming application.

In this case, $\underline{H}(\theta, \phi) \in \mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta})]$ when

$$\underline{H}(\theta, \phi) = \underline{V}_i(\underline{\beta}), \quad i = \{1, 2, \ldots, M - 1\},$$

which leads to nulls at locations that satisfy

$$H_1(\theta, \phi) = \frac{\beta_i + j}{\beta_i - j} \implies (\theta_i, \phi_i) = H_1^{-1}\left(\frac{\beta_i + j}{\beta_i - j}\right), \quad i = \{1, 2, \ldots, M - 1\}.$$

Note that, by setting $H_1(\theta, \phi) = \psi$, this result yields Equation 3.3 as stated for the two-element array in Section 3.1.

3.3.2 LENS Parameter Classification, Part 2: Generating Parameter Classes

Given the preceding relationship between $\underline{\beta}$ and $\underline{W}_{\text{LENS},M}(\underline{\beta})$, it is now necessary to develop the classification procedure that labels LENS parameters as either $\mathcal{T}$, $\mathcal{J}$, or $\overline{\mathcal{T} \cup \mathcal{J}}$. This classification is achieved by measuring the impact of β_i upon the 'closeness' of the assumed $\underline{H}_{\text{target}}$ to $\mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta})]$: (1) if β_i causes $\mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta})]$ to be close to $\underline{H}_{\text{target}}$, then β_i is considered to be target-canceling, i.e., $\beta_i \in \mathcal{T}$, (2) if β_i causes $\mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta})]$ to be far from $\underline{H}_{\text{target}}$, then β_i is considered to be jammer-canceling, i.e., $\beta_i \in \mathcal{J}$, and (3) if β_i causes $\mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta})]$ to be an intermediate

33

distance from $\underline{H}_{\text{target}}$, then β_i has an intermediate effect, i.e., $\beta_i \in \overline{\mathcal{T} \cup \mathcal{J}}$.

Distance Metric, $\Omega_i(\underline{\beta})$

One convenient distance measure between $\mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta})]$ and $\underline{H}_{\text{target}}$ is the angle $\Omega(\underline{\beta})$ between the subspace and the vector. Figure 6 shows the concept of Ω for a hypothetical subspace $\mathcal{S}$ and a vector $\underline{Z}$, where $\underline{Z}$ has been separated into two orthogonal components, $\underline{Z}_S \in \mathcal{S}$ and $\underline{Z} - \underline{Z}_S \perp \mathcal{S}$, that sum to form $\underline{Z}$. When $\Omega = 0$, then $\underline{Z}_S = \underline{Z}$, which means that $\mathcal{S}$ contains $\underline{Z}$ entirely, and the distance is 0. On the other hand, when $\Omega = 90°$, then $\underline{Z}_S = 0$, which means that $\mathcal{S}$ contains no information about $\underline{Z}$, and the distance is maximal.

In practice, the dependence of $\Omega(\underline{\beta})$ upon $\underline{\beta}$ is complicated, and it is difficult to determine the impact of an individual β_i upon this metric. It is possible, however, to define a set of metrics, $\{\Omega_i(\underline{\beta})\}$, where $\Omega_i(\underline{\beta})$ is equal to the angle between $\underline{H}_{\text{target}}$ and the span of the single null-space basis vector $\underline{V}_i(\underline{\beta})$:

$$\Omega_i(\underline{\beta}) = \cos^{-1}\left(\frac{|\underline{V}_i(\underline{\beta})^H \underline{1}|}{\|\underline{V}_i(\underline{\beta})\|_2 \, \|\underline{H}_{\text{target}}\|_2}\right) = \cos^{-1}\left(\frac{|\underline{V}_i(\underline{\beta})^H \underline{1}|}{\|\underline{V}_i(\underline{\beta})\|_2 \, \sqrt{M}}\right), \qquad (3.19)$$
$$i = \{1, 2, \ldots, M - 1\},$$

where $|\cdot|$ is the standard complex scalar magnitude, $\|\cdot\|_2$ is the Euclidean complex

vector norm , and the second equality comes from the knowledge that $\underline{H}_{target} = \underline{1}$ and that $||\underline{H}_{target}||_2 = \sqrt{M}$. Although the set $\{\Omega_i(\underline{\beta})\}$ does not provide as accurate a distance measure between $\mathcal{N}[\underline{W}_{LENS,M}(\underline{\beta})]$ and $\underline{H}_{target}$ as that provided by $\Omega(\underline{\beta})$, it does provide a general idea of this distance. Additionally, it has the advantage of being relatively simple and of measuring the effects of each LENS parameter β_i individually. Note that, like $\Omega(\underline{\beta})$, $\Omega_i(\underline{\beta})$ ranges from a minimum-distance value of $0°$ to a maximum-distance value of $90°$.

Ideal Parameter Classifications

The metric $\Omega_i(\underline{\beta})$ serves as the vehicle through which $\beta_{opt,i}$ is classified. Figure 7 shows contour plots of $\Omega_i(\underline{\beta})$, as β_i varies over the space of complex numbers, for cases of $M = 2, 3, 4,$ and 5 array elements. Specific contours indicating $\Omega_i(\underline{\beta}) = 30°, 60°, 90°$ are shown. Note the following properties of the metric $\Omega_i(\underline{\beta})$ and the information that it contains about β_i:

1. $\Omega_i(\underline{\beta})$ achieves its maximum value of 90° at precisely $M - 1$ points, known as the 'orthogonal' values of β_i: $\beta_k^\perp$, $k = \{1, 2, \ldots, M - 1\}$. At these values, the span of the individual $\underline{V}_i(\underline{\beta})$ is orthogonal to $\underline{H}_{\text{target}}$, and β_i is considered as containing jammer-canceling information only.

2. $\Omega_i(\underline{\beta}) \to 0°$ as $|\beta_i| \to \infty$. At these values, span$[\underline{V}_i(\underline{\beta})]$ practically contains $\underline{H}_{\text{target}}$, and β_i is considered as containing target-canceling information only.

3. $\Omega_i(\underline{\beta})$ decreases regularly as the distance between β_i and the orthogonal values increases. At these intermediate values, β_i contains both target-canceling and jammer-canceling information, with parameters near to $\beta_k^\perp$ being more jammer-canceling and parameters with large $|\beta_i|$ being more target-canceling.

These observations suggest the following classification of β_i:

$$
\begin{aligned}
\Omega_i(\underline{\beta}) \leq \Omega_{\mathcal{T}} &\quad\Longrightarrow\quad \beta_i \in \mathcal{T}, \\
\Omega_{\mathcal{J}} \leq \Omega_i(\underline{\beta}) &\quad\Longrightarrow\quad \beta_i \in \mathcal{J}, \\
\Omega_{\mathcal{J}} < \Omega_i(\underline{\beta}) < \Omega_{\mathcal{T}} &\quad\Longrightarrow\quad \beta_i \in \mathcal{T} \,\bar{\cup}\, \mathcal{J}.
\end{aligned}
\tag{3.20}
$$

Figure 8 indicates the three classification regions for an $M = 3$ element array when $\Omega_{\mathcal{T}} = 30°$ and $\Omega_{\mathcal{J}} = 60°$. Note that the specific values of $\Omega_{\mathcal{T}}$ and $\Omega_{\mathcal{J}}$ used in a specific LENS implementation will depend upon the array architecture and the frequency of the narrowband beamformer input. Boundary selection is discussed in more detail below for the case of ULAs.

A note on the 'orthogonal' values of β_i: It is possible to calculate explicitly the $M - 1$ orthogonal values of β_i that result in $\Omega(\beta_i) = 90°$. Specifically, these are the β_i that yield $\underline{V}_i(\underline{\beta})^H \underline{1} = 0$. Defining $\alpha_i = \frac{\beta_i + j}{\beta_i - j}$ and recalling the structure of $\underline{V}_i(\underline{\beta})$ [Equation 3.12], this requirement may be written as:

$$
\underline{V}_i(\underline{\beta})^H \underline{1} = \sum_{l=0}^{M-1} \alpha_i^l = 0.
$$

This polynomial in α_i is equal to 0 when

$$\alpha_i = e^{j\frac{2\pi k}{M}}, \quad k = 1, 2, \ldots, M - 1.$$

Solving α_i for β_i yields the orthogonal values of β_i:

$$\beta_k^\perp = \cot \frac{\pi k}{M}, \quad k = 1, 2, \ldots, M - 1. \qquad (3.21)$$

An interesting side observation here is that the $(M - 1)$-element LENS parameter vector that consists of these orthogonal values, $\underline{\beta}^\perp = [\beta_1^\perp, \beta_2^\perp, \ldots, \beta_{M-1}^\perp]^T$, yields a LENS weight vector that is equal to the uniform-weighting vector:

$$\underline{W}_{\text{LENS},M}(\underline{\beta}^\perp) = \frac{1}{M}\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T = \frac{1}{M}\underline{1}.$$

This result reinforces the choice of Ω_i as the β_i robustness metric, since the white-noise gain of $\underline{W} = \frac{1}{M}\underline{1}$ is $G_w = \frac{1}{\underline{W}^H\underline{W}} = M$ [Equation 2.15]. This is the maximum possible white-noise gain value for a weight vector that preserves $\underline{H}_{\text{target}} = \underline{1}$ with unit gain, which means that it achieves very robust beamforming.

Practical Parameter Classifications

In practice, the contours of constant $\Omega_i(\underline{\beta})$ are complicated and the preceding classification procedure can be difficult to implement. In order to yield a practical system, this research simplifies the problem by approximating the contours as boxes, centered at $\beta_i = 0$. Empirical observations demonstrate that boxes with the following proportional relationship yield reasonable matches to the desired contours:

$$\frac{\text{Height}}{\text{Width}} = \frac{|Im\{\underline{\beta}\}|}{|Re\{\underline{\beta}\}|} = \frac{\Lambda(R)}{R} = \frac{\frac{(R+4\beta_1^\perp)(R-\beta_1^\perp)}{(R+3\beta_1^\perp)}}{R}, \qquad (3.22)$$

where $\beta_1^\perp = \cot \frac{\pi}{M}$ is the largest orthogonal β_i value, as defined in Equation 3.21, and where the box half-width R lies between $\beta_1^\perp$ and ∞. Based on $\Lambda(R)$, the rectangular approximation becomes square as $R \to \infty$ and a flat line extending from real values

37

$-\beta_1^\perp = \beta_{M-1}^\perp$ to $\beta_1^\perp$ as $R \rightarrow \beta_1^\perp$. This reflects the behavior of the true contours of Ω_i, which become circular as $\Omega_i \rightarrow 0°$ and which converge to the $M - 1$ points $\beta_k^\perp$, $k = \{1, 2, \ldots, M - 1\}$, as $\Omega_i \rightarrow 90°$.

Figure 9 demonstrates the relationship between the real and approximated contours of constant $\Omega(\beta_i)$ for an $M = 3$ element array. As shown, the boxes yield reasonably good approximations to the actual contour curves, especially as Ω_i approaches $90°$. Similar results are obtained for other values of M.

Given the box-approximation to the contours of constant Ω_i, the LENS parameter classification from Equation 3.20 may be re-expressed as:

$$
\begin{aligned}
|Re\{\beta_i\}| \geq R_T \text{ OR } |Im\{\beta_i\}| \geq \Lambda(R_T) &\implies \beta_i \in \mathcal{T}, \\
|Re\{\beta_i\}| \leq R_J \text{ AND } |Im\{\beta_i\}| \leq \Lambda(R_J) &\implies \beta_i \in \mathcal{J}, \\
\left.\begin{array}{c} |Re\{\beta_i\}| > R_J \text{ OR } |Im\{\beta_i\}| > \Lambda(R_J) \\ \text{AND} \\ |Re\{\beta_i\}| < R_T \text{ AND } |Im\{\beta_i\}| < \Lambda(R_T) \end{array}\right\} &\implies \beta_i \in \overline{\mathcal{T} \cup \mathcal{J}},
\end{aligned}
\tag{3.23}
$$

which is shown graphically in Figure 10. The values of R_T and R_J yield classification boundaries that are close to those obtained using ideal contours at Ω_T and Ω_J in Figure 8. By definition, $\mathcal{T}$ and $\mathcal{J}$ should be non-overlapping, which is satisfied if $R_T > R_J$. In this case, the boundary-selection process becomes that of choosing R_T and R_J rather than that of choosing Ω_T and Ω_J. These choice are explored in more detail below for ULAs.

Special Case: Broadside ULA Architectures

This section explores the process for determining the specific values of R_T and R_J for the special case of broadside ULA systems.

Section 3.3.1 demonstrates that, for ULAs, there is a correspondence between the LENS parameters β_i and the beam-pattern null locations. Specifically, the parameter

β_i produces a null at the location (θ_i, ϕ_i) that satisfies

$$\beta_i = \cot\left(\frac{\omega\tau(\theta_i, \phi_i)}{2}\right), \qquad (3.24)$$

where $\tau(\theta, \phi)$ is the inter-element delay experience by a source from the (θ, ϕ) location. Equation 2.7 may be used to show that ULAs that are oriented broadside to the target signal have inter-element delays with the structure

$$\tau(\theta, \phi) = \frac{d}{c}\sin\theta, \qquad (3.25)$$

where θ is the angle between the source propagation direction and the plane perpendicular to the array axis [Section 2-2]. Note that $\tau(\theta, \phi)$ depends on θ only and not on ϕ, due to the cylindrically-symmetric array behavior about the array axis.

Equations 3.24 and 3.25 provide a convenient means for determining the target-canceling and jammer-canceling region boundaries that are used in the LENS classification procedure above. Specifically, a β_i that produces a null within an angle θ_T of the target location is labeled target-canceling, while a β_i that produces a null at an angle greater than θ_J from the target location is labeled jammer-canceling. Given that the target location is $\theta_T = 0$, these requirements may be expressed as:

Target-Canceling: $\quad |\theta_i - \theta_T| < \theta_T \quad \Longleftrightarrow \quad |\beta_i| > \cot\left(\frac{\omega d}{2c}\sin\theta_T\right),$

Jammer-Canceling: $\quad |\theta_i - \theta_T| > \theta_J \quad \Longleftrightarrow \quad |\beta_i| < \cot\left(\frac{\omega d}{2c}\sin\theta_J\right).$

By comparing this classification to that proposed in Equation 3.23 and by noting that β_i are assumed to be real for Equation 3.24, then this argument suggests that the desired robustness regions could be achieved using:

$$\begin{aligned} R_T &= \cot\left(\frac{\omega d}{2c}\sin\theta_T\right) \\ R_J &= \cot\left(\frac{\omega d}{2c}\sin\theta_J\right). \end{aligned} \qquad (3.26)$$

As long as $\theta_T < \theta_J$, these values will yield three distinct LENS classification regions as shown in Figure 10 and described in Equation 3.23. This research uses these

39

formulae for R_T and R_J, with $\theta_T = 15°$ and $\theta_J = 30°$.

In practice, both upper and lower limits are imposed upon R_T and R_J. The upper limit restricts any unusually large β_i values, which prevents very large values of the weight vector norm, $\underline{W}_{\text{LENS},M}(\underline{\beta})^H \underline{W}_{\text{LENS},M}(\underline{\beta})$ and prevents excessively small values of the system white noise gain $G_w = \frac{1}{\underline{W}_{\text{LENS},M}(\underline{\beta})^H \underline{W}_{\text{LENS},M}(\underline{\beta})}$ [Equation 2.15]. It must be noted that the precise relationship between $\underline{\beta}$ and the weight vector norm is complex, and so the upper bounds on R_T and R_J are not designed to restrict $\underline{W}_{\text{LENS},M}(\underline{\beta})^H \underline{W}_{\text{LENS},M}(\underline{\beta})$ to be below a specific value . This research uses $R_T < 30$ and $R_J < 10$, which have been determined empirically to yield good performance.

The lower limit on R_T and R_J ensures that the LENS parameters always has a minimal amount of 'maneuvering room' about the orthogonal parameter values $\beta_k^\perp = \cot \frac{k\pi}{M}$, $k = \{1, 2, \ldots, M-1\}$. The precise impact of these lower bounds is difficult to derive, and this research uses $R_T > \beta_1^\perp + 2$ and $R_J > \beta_1^\perp + 1$, which have been determined empirically to yield good performance.

Together, the upper and lower bounds on R_T and R_J for this research are:

$$
\begin{aligned}
\beta_1^\perp + 2 &< R_T < 30 \\
\beta_1^\perp + 1 &< R_J < 10.
\end{aligned}
\tag{3.27}
$$

3.3.3 Final Transformation: $\underline{\beta}_{\text{opt}} \rightarrow \underline{\beta}_{\text{rob}}$

Given the LENS classification procedure above, it is possible to state the actual $\underline{\beta}_{opt} \rightarrow \underline{\beta}_{rob}$ transformation, based on Equation 3.11:

$$
\beta_{rob,i} = \begin{cases} 0 & \beta_{opt,i} \in \mathcal{T} \\ \beta_{opt,i} & \beta_{opt,i} \in \mathcal{J} \\ F[\beta_{opt,i}] & \beta_{opt,i} \in \overline{\mathcal{T} \cup \mathcal{J}} \end{cases}
\tag{3.28}
$$

The first two components of this transformation are straightforward given the classes $\mathcal{T}$ and $\mathcal{J}$. For the third component, recall that $F[\beta_{opt,i}]$ should be a smooth

transition between the $\beta_{opt,i} \in \mathcal{T}$ and $\beta_{opt,i} \in \mathcal{J}$ extremes. For the box classifications of Equation 3.23 [Figure 3-7], one possible $F[\beta_{opt,i}]$ is:

$$F[\beta_{opt,i}] = \underbrace{\min\left[\frac{|Re\beta_{opt,i}| - R_{\mathcal{J}}}{R_{\mathcal{T}} - R_{\mathcal{J}}}, \frac{|Im\beta_{opt,i}| - \Lambda(R_{\mathcal{J}})}{\Lambda(R_{\mathcal{T}}) - \Lambda(R_{\mathcal{J}})}\right]}_{f[\beta_{opt,i}]} \beta_{opt,i}, \qquad (3.29)$$

where $f[\beta_{opt,i}] = 1$ for $\beta_{opt,i}$ on the border of the $\mathcal{J}$, $f[\beta_{opt,i}] = 0$ for $\beta_{opt,i}$ on the border of the $\mathcal{T}$, and $f[\beta_{opt,i}]$ linearly moves from 1 to 0 as either $Re[\beta_{opt,i}]$ progresses from $R_{\mathcal{J}}$ to $R_{\mathcal{T}}$ or $Im[\beta_{opt,i}]$ progresses from $\Lambda(R_{\mathcal{J}})$ to $\Lambda(R_{\mathcal{T}})$. Given this $F[\beta_{opt,i}]$, Equation 3.28 may also be expressed as

$$\beta_{rob,i} = \bar{f}[\beta_{opt,i}]\beta_{opt,i}, \quad \text{where } \bar{f}[\beta_{opt,i}] = \begin{cases} 0 & \beta_{opt,i} \in \mathcal{T} \\ 1 & \beta_{opt,i} \in \mathcal{J} \\ f[\beta_{opt,i}] & \beta_{opt,i} \in \overline{\mathcal{T} \cup \mathcal{J}}. \end{cases} \qquad (3.30)$$

3.3.4 Example Cases

In order to gain some understanding into how the LENS robustness restriction actually operates, this section presents some simple examples based upon a two-element array with an element spacing of $d = 0.07$ m. This array is oriented broadside to the target location, which means that no steering delays are needed to make $\underline{H}_{target} = \underline{1}$. Assuming far-field sources, then Equations 2.6 and 2.7 may be combined to yield the source-to-array transfer function expression:

$$\underline{H}(\theta, \phi) = \begin{bmatrix} 1 \\ e^{j\omega \frac{d}{c} \sin\theta} \end{bmatrix} = \begin{bmatrix} 1 \\ e^{j2\pi f \frac{0.07\,m}{345\,m/sec} \sin\theta} \end{bmatrix},$$

where an acoustic propagation velocity of $c = 345$ m/sec has been used and where θ is the angle between the source propagation direction and the plane that is perpendicular to the array axis [Section 2.2.1]. In the azimuth plane, θ may be interpreted as the angle to the left of the assumed target location.

Given this array, consider the following four source situations, where a single

41

source, with source-to-array transfer function $\underline{H}^k$, dominates $\beta^k_{opt,1}$. The values of $\beta^k_{opt,1}$ that yield $\underline{W}_{\text{LENS},M}(\beta^k_{opt,1})^H \underline{H}^k = 0$ are also given, as determined from Equation 3.3. Note that a perfectly-aligned target source, with $\underline{H}(\theta,\phi) = \underline{H}_{\text{target}}$, is not considered, since, by definition, $\underline{W}_{\text{LENS},2}(\beta_1)$ cannot null this location.

1. $f = 500\,\text{Hz}$, $10°$ mis-aligned target, matched array elements:

$$\underline{H}^1 = \begin{bmatrix} 1 \\ e^{j2\pi 500\,\text{Hz}\frac{0.07\,\text{m}}{345\,\text{m/sec}}\sin 10°} \end{bmatrix} \quad \Longleftrightarrow \quad \beta^1_{opt,1} = 18.01.$$

2. $f = 500\,\text{Hz}$, aligned target, mis-matched array elements:

$$\underline{H}^2 = \begin{bmatrix} 1 \\ 0.9 \end{bmatrix} \quad \Longleftrightarrow \quad \beta^2_{opt,1} = -j19.0.$$

3. $f = 500\,\text{Hz}$, $45°$ jammer, matched array elements:

$$\underline{H}^3 = \begin{bmatrix} 1 \\ e^{j2\pi 500\,\text{Hz}\frac{0.07\,\text{m}}{345\,\text{m/sec}}\sin 45°} \end{bmatrix} \quad \Longleftrightarrow \quad \beta^3_{opt,1} = 4.36.$$

4. $f = 500\,\text{Hz}$, $45°$ jammer, mis-matched array elements:

$$\underline{H}^4 = \begin{bmatrix} 1 \\ 0.9e^{j2\pi 500\,\text{Hz}\frac{0.07\,\text{m}}{345\,\text{m/sec}}\sin 45°} \end{bmatrix} \quad \Longleftrightarrow \quad \beta^4_{opt,1} = 4.13 - j1.0.$$

Figure 11 plots these four $\beta^k_{opt,1}$ values as well as the LENS robustness regions for $f = 500\,\text{Hz}$, defined according to the R_T and R_J of Equation 3.26 with $\theta_T = 15°$ and $\theta_J = 30°$.

These plots demonstrate the LENS robustness restriction at work. The target-canceling LENS parameters, $\beta^1_{opt,1}$ and $\beta^2_{opt,1}$, lie in the T region, which means that the robustness restriction transforms them to the neutral value of $\beta^1_{rob,1} = \beta^2_{rob,1} = 0$. On the other hand, the jammer-canceling LENS parameters, $\beta^3_{opt,1}$ and $\beta^4_{opt,1}$, lie within

4   The Overall LENS Algorithm

Figure 12 summarizes the overall LENS algorithm.   Since this
research will study ULAs that have been oriented broadside to the target location,
the robustness transformation boundaries derived in Section 3.3.2 will be used:

$$R_T = \cot\left(\frac{\omega d}{2c}\sin 15°\right), \qquad \text{subject to} \qquad \beta_1^\perp + 2 < R_T < 30, \qquad (3.31)$$
$$R_J = \cot\left(\frac{\omega d}{2c}\sin 30°\right), \qquad\qquad\qquad \beta_1^\perp + 1 < R_J < 10,$$

where d is the inter-element array spacing, c is the propagation velocity, and $\beta_1^\perp$ is
the largest magnitude orthogonal value [Equation 3.21].

One LENS feature is immediately obvious when comparing LENS to traditional
CMV processing from Section 2.3.2: LENS separates the robustness constraining pro-
cess from the beamforming power minimization, while CMV combines the robustness
constraints into the beamforming power minimization. It is the separation of these
two components that allows LENS processing to exhibit direct and flexible control
over the beamforming.

5   Modified LENS: Skew LENS Processing

Skew LENS processing is a relatively simple modification of standard LENS processing. Specifically, it consists of the following four steps:

1. Given the standard LENS input, which has been steered to have $\underline{H}_{target} = \underline{1}$, form the skewed input signal $\underline{X}_\zeta = Z\underline{X}$, where:

$$Z = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \zeta & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \zeta^{M-1} \end{bmatrix}. \tag{5.1}$$

The matrix Z is the skewing matrix and ζ is the skewing parameter.

2. Generate the optimal skewed LENS parameter set $\underline{\beta}_{\zeta,opt}$ based on $\underline{X}_\zeta$, i.e., use the standard LENS definition [Equation 3.8], with $R_{X_\zeta X_\zeta}$ rather than R_{XX}.

44

3. Restrict $\underline{\beta}_{\zeta,opt}$ for robustness to form $\underline{\beta}_{\zeta,rob}$.

4. Form the weight vector $\underline{W}_{LENS,M}(\underline{\beta}_{\zeta,rob})$ and use this to form the beamformer output $Y = \underline{W}_{LENS,M}(\underline{\beta}_{\zeta,rob})^H \underline{X}_\zeta$.

Given this description, it is clear that Skew LENS reduces to standard LENS processing when $\zeta = 1$ and Z becomes the identity matrix

It is very important to note that the optimal Skew LENS parameter set, $\underline{\beta}_{\zeta,opt}$, is defined using the standard LENS definition [Equation 3.8] applied to $\underline{X}_\zeta$, i.e., $\underline{\beta}_{\zeta,opt}$ is designed to preserve sources with $\underline{H}(\theta,\phi) = \underline{1}$. Given the Skew LENS definition, however, it is clear that the assumed target location now has a transfer function of

$$\underline{H}_{\zeta,\text{target}} = Z\underline{1} = \begin{bmatrix} 1 \\ \zeta \\ \vdots \\ \zeta^{M-1} \end{bmatrix}, \tag{5.2}$$

which means that $\underline{\beta}_{\zeta,opt}$ may actually cancel the assumed target location! As described below, however, the introduction of this controlled and very structured error into $\underline{H}_{\text{target}}$ concentrates target-canceling beamforming information into a single LENS parameter, which can improve the overall LENS performance.

The following sections present the logic behind Skew LENS processing, discuss the selection of the skewing parameter ζ, develop a modified LENS robustness restriction for Skew LENS processing, and present complete Skew LENS description.

5.1 Skew LENS Basics

For the special case of idealized narrowband far-field sources with a mis-aligned target,

LENS algorithm can completely separate the target-canceling and jammer-canceling beamforming effects within the optimal LENS parameter set. This allows LENS to prevent target cancellation without limiting jammer cancellation, which results in array gain levels that are nearly-equal to those observed with a perfectly-aligned target.

For all other array environments, LENS cannot completely separate target-canceling from jammer-canceling information in the optimal LENS parameter set. This means that LENS cannot restrict target cancellation without simultaneously limiting jammer cancellation, which prevents LENS from maintaining near-ideal performance levels. Skew LENS processing is an attempt to increase the separation of target-canceling and jammer-canceling information in the optimal LENS parameter set for all array environments and for all sources of target-cancellation.

Skew LENS processing achieves this goal through the skewing matrix Z. Specifically, Z concentrates the possible target-canceling LENS parameter values into a relatively small subset of the overall LENS parameter space. By concentrating the range of possible target-canceling LENS parameter values, Skew LENS processing allows these values to be covered by a single LENS parameter. This frees the remaining parameters for jammer cancellation and increases the separation of target-canceling and jammer-canceling information within $\underline{\beta}_{\zeta,opt}$.

To see how Z concentrates target-canceling LENS parameters, recall the behavior of these parameters. Section 3.3.1 shows that $\underline{\beta}$ describes a basis, $\{\underline{V}_i(\underline{\beta})\}$, for the null-space $\mathcal{N}[\underline{W}_{LENS,M}(\underline{\beta})]$, of $\underline{W}_{LENS,M}(\underline{\beta})$, where

$$
\underline{V}_i(\underline{\beta}) = \begin{bmatrix} 1 \\ \left(\frac{\beta_i+j}{\beta_i-j}\right) \\ \left(\frac{\beta_i+j}{\beta_i-j}\right)^2 \\ \vdots \\ \left(\frac{\beta_i+j}{\beta_i-j}\right)^{M-1} \end{bmatrix}, \quad i = \{1, 2, \ldots, M-1\}.
$$

A target-canceling LENS parameter, β_i, corresponds to a $\underline{V}_i(\underline{\beta})$ that is close to the $\underline{H}_{target}$. Apply this observation to standard and Skew LENS processing:

46

1. For standard LENS, $[\underline{H}_{target} = \underline{1}]$, this closeness occurs when

$$|\beta_i| \to \infty.$$

2. For Skew LENS, $[\underline{H}_{\zeta,target} = Z\underline{1}]$ this closeness occurs when

$$\zeta = \frac{\beta_i + j}{\beta_i - j} \quad \Rightarrow \quad \beta_i = j\frac{\zeta + 1}{\zeta - 1},$$

assuming $\zeta \neq 1$.

Now, assume that the actual target-to-array transfer function, $\underline{\hat{H}}_{target}$, is near to, but not equal to, $\underline{H}_{target} = \underline{1}$. For standard LENS, as $\underline{\hat{H}}_{target}$ varies from $\underline{H}_{target}$, it is clear that that the target-canceling LENS parameters will cover the entire large-$|\beta_i|$ region. For Skew LENS, on the other hand, as $\underline{\hat{H}}_{\zeta,target} = Z\underline{\hat{H}}_{target}$ varies from $\underline{H}_{\zeta,target}$, it is clear that the target-canceling LENS parameters will be clustered about $\beta_i = j\frac{\zeta+1}{\zeta-1}$. This demonstrates that, as desired, Skew LENS concentrates the target-canceling LENS parameters.

5.2 Selecting the Skew Parameter, ζ

The Skew LENS parameter, ζ, determines the amount of concentration that Z introduces into the target-canceling LENS-parameter values and the overall ability of Skew LENS to separate target-canceling from jammer-canceling information. In general, as ζ deviates further from the non-skewed case of $\zeta = 1$, more skewing and more concentration occur. Care must be taken, however, when selecting ζ, and the following rules should be observed:

1. ζ should not yield $Z\underline{H}(\theta, \phi) = \underline{1}$ for any non-target (θ, ϕ):
   Since that $\underline{\beta}_{\zeta,opt}$ is defined using the standard LENS optimization, it preserves sources with source-to-array transfer functions of $Z\underline{H}(\theta, \phi) = \underline{1}$. Given the beamforming goal of jammer signal attenuation, it is clearly undesirable for Z to map a jammer source to have $Z\underline{H}(\theta, \phi) = \underline{1}$.

Consider as an example broadside ULAs. As shown in Equation 2.6, these arrays have non-skewed source-to-array transfer functions of the form:

$$\underline{H}(\theta,\phi) = \begin{bmatrix} 1 \\ e^{j\omega\tau_1(\theta,\phi)} \\ \vdots \\ e^{j\omega(M-1)\tau_1(\theta,\phi)} \end{bmatrix},$$

where $\tau(\theta,\phi) = \frac{d}{c}\sin\theta$, d is the inter-element spacing and c is the propagation velocity. Given this $\underline{H}(\theta,\phi)$, a skewing factor of $\zeta = e^{-j\omega\tau(\theta_1,\phi_1)}$ would yield $Z\underline{H}(\theta_1,\phi_1) = \underline{1}$, and that Skew LENS would preserve sources from the (θ_1,ϕ_1) location! For ULAs, $Z\underline{H}(\theta,\phi) = \underline{1}$ is avoided when $|\zeta| \neq 1$.

2. ζ should not skew the processing excessively far from standard LENS processing. i.e., $|\zeta - 1|$ should not be too large:

As ζ deviates further from the non-skewed $\zeta = 1$, the concentration of the target-canceling LENS parameter values increases. Unfortunately, if ζ deviates too far, then the skewing also begins to concentrate the jammer-canceling information into the same set of LENS parameter values. This reduces the separation of the target-canceling and jammer-canceling information.

Consider as an example the broadside ULA situation from above, which has the non-skewed source-to-array transfer functions given above. At low frequencies, the delay term $\tau(\theta,\phi)$ is small for both target and jammer locations, which means that $\underline{H}(\theta,\phi)$ is always 'near' to $\underline{H}_{target} = \underline{1}$. A skewing parameter that deviates largely from $\zeta = 1$ masks entirely the $e^{j\omega\tau(\theta,\phi)}$ term in $Z\underline{H}(\theta,\phi)$, which means that the jammer-canceling LENS parameter values for this location are concentrated together with the target-canceling LENS parameter values. Empirical observations conducted for this research suggest that $|\zeta - 1| < 0.05$ avoids excessive concentration of jammer-canceling LENS parameter values.

For this research, Skew LENS is implemented with $\zeta = 0.95$.

5.3 Skew LENS Robustness Restriction

As described above, Skew LENS processing applies the skewing matrix Z to the standard LENS beamformer input, $\underline{X}$, in order to concentrate the possible target-canceling LENS parameter values into a relatively small subset of the LENS parameter space. Since this transformation re-maps the target-canceling and jammer-canceling LENS parameter regions, it is necessary to re-define the LENS robustness regions $\mathcal{T}$, $\mathcal{J}$, and $\overline{\mathcal{T} \cup \mathcal{J}}$ (target-canceling, jammer-canceling, and intermediate, respectively) that are used in the LENS robustness restriction of Section 3.3.

Such a re-definition would be a time-consuming and rather difficult procedure, but, fortunately, there is a relatively simple means of by-passing this problem. It is possible to convert $\underline{\beta}_{\zeta,opt}$ into the non-skewed LENS parameter set $\underline{\beta}_{ns,opt}$, for which the target-canceling and jammer-canceling LENS parameter regions are mapped back to their original, non-skewed, standard-LENS locations. This $\underline{\beta}_{ns,opt}$ may then be restricted for robustness using the standard LENS procedure of Section 3.3 to yield $\underline{\beta}_{ns,rob}$, and this result may be 're-skewed' to form $\underline{\beta}_{\zeta,rob}$.

At first glance, it may appear that this non-skewing transformation negates all of the beneficial effects of Skew LENS processing and that $\underline{\beta}_{ns,opt}$ may simply equal the original, standard LENS parameter $\underline{\beta}_{opt}$. This behavior is not, in fact, true – all beneficial effects of Skew LENS processing arise out of the actual power minimization that yields $\underline{\beta}_{\zeta,opt}$, and the non-skewing transformation is simply a tool that assists in the robustness restriction to form $\underline{\beta}_{\zeta,rob}$.

5.3.1 Relationship between $\underline{\beta}_{\zeta}$ and the Non-Skewed $\underline{\beta}_{ns}$

A Skew LENS parameter set $\underline{\beta}_{\zeta}$ and its non-skewed counterpart $\underline{\beta}_{ns}$ are considered equivalent if they describe the same beamforming behavior. Specifically, these two parameter sets will be considered equivalent when the beamforming accomplished by

$$\underline{W}_{\text{LENS},M}(\underline{\beta}_{\zeta})^H \underline{X}_{\zeta} = \underline{W}_{\text{LENS},M}(\underline{\beta}_{\zeta})^H Z \underline{X}$$

is equivalent to the beamforming accomplished by

$$\underline{W}_{\text{LENS},M}(\underline{\beta}_{\text{ns}})\underline{X},$$

where Z is the skewing matrix of Equation 5.1 and the relation $\underline{X}_\zeta = Z\underline{X}$ has been used. Such beamforming equivalence is attained when

$$\mathcal{N}[Z^H\underline{W}_{\text{LENS},M}(\underline{\beta}_\zeta)] = \mathcal{N}[\underline{W}_{\text{LENS},M}(\underline{\beta}_{\text{ns}})] \tag{5.3}$$

Equation 5.3 is true when $\underline{\beta}_\zeta$ and $\underline{\beta}_{\text{ns}}$ are related using the following transformation pair:

$$
\begin{aligned}
\text{Skewed} \Rightarrow \text{Non-Skewed} \quad & \beta_{\text{ns},i} = j\frac{(\beta\zeta,i+j)+\zeta(\beta_{\zeta,i}+j)}{(\beta_{\zeta,i}+j)-\zeta(\beta_{\zeta,i}+j)} \\
\text{Non-Skewed} \Rightarrow \text{Skewed} \quad & \beta_{\zeta,i} = j\frac{\zeta(\beta_{\text{ns},i}+j)+(\beta_{\text{ns},i}+j)}{\zeta(\beta_{\text{ns},i}+j)-(\beta_{\text{ns},i}+j)}
\end{aligned}
\tag{5.4}
$$

5.3.2 Skew LENS Robustness Restriction Summary

Figure 13 presents the overall robustness restriction process for Skew LENS processing. It consists of the standard LENS robustness restriction nested between the $\underline{\beta}_{\zeta,opt} \rightarrow \underline{\beta}_{\text{ns},opt}$ and $\underline{\beta}_{\text{ns},rob} \rightarrow \underline{\beta}_{\zeta,rob}$ transformations.

5.4 Overall Skew LENS Algorithm

Figure 14 shows the overall Skew LENS flow diagram. This resembles the standard LENS flow diagram from Section 3.4, but it adds the input-skewing step as Step 1 and modifies the LENS robustness restriction as described by Figure 13.

6 LENS Implementation

The main computational component of either standard LENS [section 3] or Skew LENS [section 5] processing is the determination of the optimal LENS parameter set, $\underline{\beta}_{opt}$ or $\underline{\beta}_{\zeta,opt}$. Once this optimal parameter set has been determined, it is a straightforward process to restrict the parameters for robustness and to generate the final beamforming weight vector. This chapter develops and simulates two implementation methods that use the temporal stream of narrowband array input observations, $\{\underline{X}^{(l)}\}$ to form a running estimate $\underline{\beta}_{opt}^{(l)}$ of the optimal LENS parameter set. Note that, although this chapter uses the terminology of standard LENS processing, the identical methods may be used to obtain $\underline{\beta}_{\zeta,opt}^{(l)}$ from $\{\underline{X}_{\zeta}^{(l)}\}$ for Skew LENS processing.

Recall from section 3 that the optimal LENS parameter set is defined as the parameter set that minimizes the expected beamformer output power that arises when using the LENS weight vector $\underline{W}_{LENS,M}(\underline{\beta})$:

$$\underline{\beta}_{opt} = \arg \min_{\underline{\beta} \in \mathbb{C}^{M-1}} \underbrace{\underline{W}_{LENS,M}(\underline{\beta})^H R_{XX} \underline{W}_{LENS,M}(\underline{\beta})}_{\stackrel{\text{def}}{=} P[\beta]}, \qquad (6.1)$$

where $R_{XX} = E[\underline{X}\underline{X}^H]$ is the input covariance matrix and $P[\beta]$ is defined to be the expected beamformer output power. As shown in Section 3.2.1 [Equation 3.10], the

closed-form solution to this optimization may be written as:

$$\underline{\beta}_{opt} = -\text{roots}\left(\frac{\underline{1}^H R_{XX}^{-1} Q_M^H}{\underline{1}^H R_{XX}^{-1} \underline{1}}\right),\tag{6.2}$$

where the operation roots($\underline{p}$) returns the set of roots for the polynomial with coefficients given by the elements of the vector $\underline{p}$.

Since R_{XX} is unknown and time-varying, Equation 6.2 generally cannot be used to generate a single $\underline{\beta}_{opt}$ that describes the entire input. Instead, $\underline{\beta}_{opt}$ must be estimated actively from the temporal stream of input observations, $\{\underline{X}^{(l)}\}$. The most obvious LENS implementation would use $\{\underline{X}^{(l)}\}$ to formulate a time-dependent input covariance estimate $\hat{R}_{X'X}^{(l)}$, which, through Equation 6.2, would yield a time-dependent optimal parameter set $\underline{\beta}_{opt}^{(l)}$. Unfortunately, the matrix inversion and polynomial root calculation render this system computationally expensive in practice. This chapter develops two alternative, iterative solutions to the LENS optimization of Equation 6.1 and uses these solutions to develop actual LENS implementations.

Relaxation Approximations

This class of adaptive solution arises out of the Gauss-Seidel relaxation-iteration for linear systems of equations   and solves Equation 6.1 by cycling through the LENS parameters and minimizing $P[\underline{\beta}]$ with respect to each β_i individually while holding the remaining β_m, $m \neq i$, fixed to their most recent values.

Gradient Approximations

This class of adaptive solution calculates the gradient of $P[\underline{\beta}]$ with respect to the individual β_i parameters and follows the gradient to minimize $P[\underline{\beta}]$.

6.1   Useful Formulae: $\frac{\partial P[\underline{\beta}]}{\partial \beta_{R,i}}$, $\frac{\partial P[\underline{\beta}]}{\partial \beta_{I,i}}$, $\frac{\partial^2 P[\underline{\beta}]}{\partial \beta_{R,i}^2}$, and $\frac{\partial^2 P[\underline{\beta}]}{\partial \beta_{I,i}^2}$.

Before developing the Relaxation and Gradient LENS implementations, this section presents expressions for the partial derivatives $\frac{\partial P[\underline{\beta}]}{\partial \beta_{R,i}}$, $\frac{\partial P[\underline{\beta}]}{\partial \beta_{I,i}}$, $\frac{\partial^2 P[\underline{\beta}]}{\partial \beta_{R,i}^2}$, and $\frac{\partial^2 P[\underline{\beta}]}{\partial \beta_{I,i}^2}$, where $\beta_i = \beta_{R,i} + j\beta_{I,i}$. These partial derivatives are required for the development of both iterative LENS solutions.

Given

$$\frac{\partial P[\beta]}{\partial \beta_{R,i}} = 2B_{R,i} + 2\beta_{R,i}C_i,$$

$$\frac{\partial P[\beta]}{\partial \beta_{I,i}} = -2B_{I,i} + 2\beta_{I,i}C_i,$$

$$\frac{\partial^2 P[\beta]}{\partial \beta_{R,i}^2} = C_i,$$

$$\frac{\partial^2 P[\beta]}{\partial \beta_{I,i}^2} = C_i,$$

(6.3)

where the parameters B_i and C_i are given by:

$$B_i = \tfrac{1}{4} \begin{bmatrix} 0 & \underline{U}_{\text{LENS},M,i}(\underline{\beta})^H \end{bmatrix} T_M^H R_{XX} T_M \begin{bmatrix} \underline{U}_{\text{LENS},M,i}(\underline{\beta}) \\ 0 \end{bmatrix},$$

$$C_i = \tfrac{1}{4} \begin{bmatrix} 0 & \underline{U}_{\text{LENS},M,i}(\underline{\beta})^H \end{bmatrix} T_M^H R_{XX} T_M \begin{bmatrix} 0 \\ \underline{U}_{\text{LENS},M,i}(\underline{\beta}) \end{bmatrix}.$$

(6.4)

In this definition: (1) T_M is the transformation matrix from Equation 3.7 that transforms the LENS beamforming weight vector $\underline{W}_{\text{LENS},M}(\underline{\beta})$ into the alternative beamforming weight vector $\underline{U}_{\text{LENS},M}(\underline{\beta})$ [Appendix B] and (2) $\underline{U}_{\text{LENS},M,i}(\underline{\beta})$ is a vector that is entirely independent of β_i and is defined to be:

$$\underline{U}_{\text{LENS},M,i}(\underline{\beta})^H = \left(\frac{1}{2}\right)^{M-2} [\, 1 \quad \beta_1 \,] * \cdots * [\, 1 \quad \beta_{i-1} \,] *$$
$$[\, 1 \quad \beta_{i+1} \,] * \cdots * [\, 1 \quad \beta_{M-1} \,].$$

(6.5)

Note that $\underline{U}_{\text{LENS},M}(\underline{\beta})$ [Equation 3.6] and $\underline{U}_{\text{LENS},M,i}$ are related through β_i:

$$\underline{U}_{\text{LENS},M}(\underline{\beta})^H = \frac{1}{2} \begin{bmatrix} \underline{U}_{\text{LENS},M,i}(\underline{\beta})^H & 0 \end{bmatrix} + \frac{\beta_i}{2} \begin{bmatrix} 0 & \underline{U}_{\text{LENS},M,i}(\underline{\beta})^H \end{bmatrix}.$$ (6.6)

6.2 Relaxation LENS Implementation

Relaxation LENS processing solves for $\underline{\beta}_{opt}$ iteratively by cycling through the LENS parameters one-by-one and minimizing Equation 6.1 with respect to each β_i individually, while holding the remaining parameters fixed. This process is based upon the idea of the Gauss-Seidel iteration to solve linear systems of equations.

53

In order to illustrate how this relaxation iteration operates, this section begins by exploring Equation 6.1 in the case of a known and fixed input covariance matrix R_{XX}. It then generalizes the result to yield the Relaxation LENS implementations that combine the relaxation iteration with a time-dependent input-covariance estimate $\hat{R}_{XX}^{(l)}$ to generate a time-dependent optimal LENS parameter set estimate $\underline{\beta}_{opt}^{(l)}$.

6.2.1   Covariance Relaxation

Recall the basic LENS problem from Equation 6.1:

$$\underline{\beta}_{opt} = \arg \min_{\underline{\beta} \in \mathbf{C}^{M-1}} P[\underline{\beta}].$$

The relaxation LENS solution to this minimization may be stated as follows.

1. Initialize the process with an arbitrary parameter vector $\hat{\underline{\beta}}_{opt}[0]$, such as the orthogonal LENS parameters $\underline{\beta}^{\perp}$ [Equation 3.21].

2. Update $\hat{\underline{\beta}}_{opt}[k]$ into $\hat{\underline{\beta}}_{opt}[k+1]$ by cycling through the LENS parameters β_i exactly once and minimizing $P[\underline{\beta}]$ with respect to each β_i while holding the remaining β_m, $m \neq i$, fixed at their most recent values:

$$\hat{\beta}_{opt,i}[k+1] = \arg \min_{\beta_i \in \mathbf{C}} P[\underline{\beta}]\Big|_{\beta_m, \, m \, \neq \, i \text{ fixed as desired}}. \qquad (6.7)$$

Consider this update process in detail for an arbitrary β_i. Since updates have already been performed for β_m, $m < i$, the most recent values of β_m, $m \neq i$ are:

$$\beta_m = \begin{cases} \hat{\beta}_{opt,m}[k+1] & m < i, \\ \hat{\beta}_{opt,m}[k] & m > i. \end{cases}$$

Given these fixed β_m, $m \neq i$, it is possible to form B_i and C_i [Equation 6.4] and to use these to define $\frac{\partial P[\underline{\beta}]}{\partial \beta_{R,i}}$ and $\frac{\partial P[\underline{\beta}]}{\partial \beta_{I,i}}$ [Equation 6.3]. Since

$$\frac{\partial^2 P[\underline{\beta}]}{\partial \beta_{R,i}^2} = \frac{\partial^2 P[\underline{\beta}]}{\partial \beta_{I,i}^2} = C_i \geq 0,$$

54

then the solution to Equation 6.7 is obtained by setting $\frac{\partial P[\beta]}{\partial \beta_{R,i}}$ and $\frac{\partial P[\beta]}{\partial \beta_{I,i}}$ to 0 and solving:

$$\left. \begin{array}{lll} \frac{\partial P[\beta]}{\partial \beta_{R,i}} = 2B_{R,i} + 2\beta_{R,i}C_i = 0 & \Rightarrow & \beta_{R,i} = -\frac{B_{R,i}}{C_i}, \\ \frac{\partial P[\beta]}{\partial \beta_{I,i}} = -2B_{I,i} + 2\beta_{I,i}C_i = 0 & \Rightarrow & \beta_{I,i} = \frac{B_{I,i}}{C_i}, \end{array} \right\} \Rightarrow \beta_i = -\frac{B_i^*}{C_i}.$$

This yields $\beta_{opt,i}[k+1] = -\frac{B_i^*}{C_i}$ to solve Equation 6.7.

As $k \to \infty$, $\hat{\underline{\beta}}_{opt}[k]$ converges to the true underlying $\underline{\beta}_{opt}$ [as demonstrated by the simulations in Section 6.5.1]. Figure 15 depicts the overall LENS relaxation iteration process.

6.2.2 Relaxation LENS Processing

Now that the basic concept of the LENS relaxation iteration in the case of a known and fixed input covariance has been derived, it is necessary to modify the iteration to suit the more realistic situation in which the input covariance is unknown and time-varying. This is achieved by using the temporal stream of array inputs, $\{\underline{X}^{(l)}\}$, to form a time-dependent estimate $\hat{R}_{XX}^{(l)}$ of R_{XX}.

The most direct Relaxation LENS implementation would use each $\hat{R}_{XX}^{(l)}$ to form a corresponding $P[\beta]^{(l)}$ and would apply a full relaxation solution until $\hat{\underline{\beta}}_{opt}^{(l)}$ has been generated to the desired precision. Assuming that the relaxation iteration converges to the true $\underline{\beta}_{opt}$ solution to Equation 6.1 (this is demonstrated below), this system is equivalent to one that uses Equation 6.2 directly to generate $\hat{\underline{\beta}}_{opt}^{(l)}$ from $\hat{R}_{XX}^{(l)}$. Unfortunately, this implementation has the disadvantage that it requires several iterations to obtain each $\hat{\underline{\beta}}_{opt}^{(l)}$, which may be computationally demanding.

Simplified adaptive relaxation implementations result by realizing that $\hat{R}_{XX}^{(l)}$ is correlated with $\hat{R}_{XX}^{(l-1)}$:   many of the same input observations go into the averages that form $\hat{R}_{XX}^{(l)}$ and $\hat{R}_{XX}^{(l-1)}$. For this reason, $\hat{\underline{\beta}}_{opt}^{(l)}$ and $\hat{\underline{\beta}}_{opt}^{(l-1)}$ are generally 'close' to one another. The number of relaxation iterations required to obtain $\hat{\underline{\beta}}_{opt}^{(l)}$ can be substantially reduced by using $\hat{\underline{\beta}}_{opt}^{(l-1)}$ as the initialization for the iteration to solve for $\hat{\underline{\beta}}_{opt}^{(l)}$. In fact, if $\hat{\underline{\beta}}_{opt}^{(l)}$ is sufficiently slowly-changing, then it is

possible to update only a fraction of the LENS parameters at each input step while maintaining reasonable solution accuracy.

Figure 16 shows two possible simplified implementations, using the notation $\text{Rlx}[\hat{\beta}_{opt,i}^{(l)}]$ to denote a relaxation iteration update of the parameter $\hat{\beta}_{opt,i}^{(l)}$.

1. The <u>Complete Update</u> system updates each of the $M - 1$ LENS parameters in $\underline{\hat{\beta}}_{opt}^{(l-1)}$ to yield $\underline{\hat{\beta}}_{opt}^{(l)}$.

2. The <u>Partial Update</u> system updates exactly one of the LENS parameters in $\underline{\hat{\beta}}_{opt}^{(l-1)}$ to yield $\underline{\hat{\beta}}_{opt}^{(l)}$. Note that for this system, care must be taken so that the relaxation update still cycles through all $M - 1$ LENS parameters. This cycling is achieved by updating parameter $i = [1 + (l \bmod M - 1)]$ at input step l.

The primary trade-off between these two implementations is one of implementation complexity versus solution accuracy and adaptation speed. By updating only one parameter per input observation, the partial update implementation reduces the computations required by a factor of $M - 1$ relative to the complete update implementation. This reduced complexity comes at the cost of updating each LENS parameter only once every $M - 1$ input observations, which increases the system adaptation time to a changing environment and reduces the overall solution accuracy. As shown by the simulations of Section 6.5.2, however, the covariance estimate $\hat{R}_{XX}^{(l)}$ generally changes slowly enough that this latter effect is minor and that Complete and Partial Update Relaxation LENS implementations yield nearly identical results.

6.3   Gradient LENS Implementation

Gradient LENS processing takes a different approach to solving Equation 6.1. Specifically, it calculates the gradient of $P[\beta]$ with respect to the LENS parameters and steps along this gradient until $P[\beta]$ has been minimized. In a manner similar to that of the previous section, this section develops gradient LENS processing by considering the problem in the case of a known covariance matrix, and then generalizing the

56

result to yield the Gradient LENS implementation that generates the time-dependent $\hat{\beta}_{opt}^{(l)}$ for an unknown covariance matrix.

6.3.1 Known Covariance Gradient Search

Recall the basic LENS parameter problem from Equation 6.1,

$$\underline{\beta}_{opt} = \arg \min_{\underline{\beta} \in \mathbb{C}^{M-1}} P[\underline{\beta}].$$

The basic gradient-based LENS solution to this minimization may be stated as follows.

1. Initialize $\hat{\underline{\beta}}_{opt}[0]$ with an arbitrary parameter vector, such as the orthogonal LENS parameter set, $\underline{\beta}^{\perp}$, from Equation 3.21.

2. Update $\hat{\underline{\beta}}_{opt}[k]$ into $\hat{\underline{\beta}}_{opt}[k+1]$ by calculating the gradient of $P[\underline{\beta}]$ at $\hat{\underline{\beta}}_{opt}[k]$ and taking a small step along this gradient toward the function minimum:

$$\hat{\underline{\beta}}_{opt}[k+1] = \hat{\underline{\beta}}_{opt}[k] - \mu \nabla P[\underline{\beta}]\Big|_{\underline{\beta} = \hat{\underline{\beta}}_{opt}[k]}, \tag{6.8}$$

where Equation 6.3 allows $\nabla P[\underline{\beta}]$ to be written as:

$$\nabla P[\underline{\beta}] = \begin{bmatrix} \frac{\partial P[\beta]}{\partial \beta_{R,1}} \\ \frac{\partial P[\beta]}{\partial \beta_{R,2}} \\ \vdots \\ \frac{\partial P[\beta]}{\partial \beta_{R,M-1}} \end{bmatrix} + j \begin{bmatrix} \frac{\partial P[\beta]}{\partial \beta_{I,1}} \\ \frac{\partial P[\beta]}{\partial \beta_{I,2}} \\ \vdots \\ \frac{\partial P[\beta]}{\partial \beta_{I,M-1}} \end{bmatrix} = 2 \begin{bmatrix} B_1^* + \beta_1 C_1 \\ B_2^* + \beta_2 C_2 \\ \vdots \\ B_{M-1}^* + \beta_{M-1} C_{M-1} \end{bmatrix}.$$

Using the definitions of B_i and C_i [Equation 6.4], the vectors $\underline{U}_{\text{LENS},M}(\underline{\beta})$ and $\underline{U}_{\text{LENS},M,i}(\underline{\beta})$ [Equations 3.6 and 6.5], and the relationship between $\underline{U}_{\text{LENS},M}(\underline{\beta})$ and $\underline{U}_{\text{LENS},M,i}(\underline{\beta})$ [Equation 6.6], $B_i^* + \beta_i C_i$ may be re-written as:

$$\begin{aligned}
B_i^* + \beta_i C_i &= \frac{1}{2} \underline{U}_{\text{LENS},M}(\underline{\beta})^H R_{X_T X_T} \begin{bmatrix} 0 \\ \underline{U}_{\text{LENS},M,i}(\underline{\beta})^H \end{bmatrix}, \\
&= \frac{1}{2} \begin{bmatrix} 0 & \underline{U}_{\text{LENS},M,i}(\underline{\beta})^T \end{bmatrix} R_{X_T X_T}^* \underline{U}_{\text{LENS},M}(\underline{\beta})^*,
\end{aligned}$$

where the second equality arises from the fact that $(B_i^* + \beta_i C_i)^T = B_i^* + \beta_i C_i$. This expression for $B_i^* + \beta_i C_i$ yields:

$$\nabla P[\underline{\beta}] = \underbrace{\begin{bmatrix} 0 & \underline{U}_{LENS,M,1}(\underline{\beta})^T \\ 0 & \underline{U}_{LENS,M,2}(\underline{\beta})^T \\ \vdots & \vdots \\ 0 & \underline{U}_{LENS,M,M-1}(\underline{\beta})^T \end{bmatrix}}_{T_{LENS,M}(\underline{\beta})} R_{X_T X_T}^* \underline{U}_{LENS,M}(\underline{\beta})^*. \qquad (6.9)$$

5   The step-size parameter μ in the gradient update [Equation 6.8] governs the overall rate and quality of the iteration convergence. A larger step-size results in faster initial convergence towards the optimum, but exhibits oscillation about the optimal value (also known as misadjustment). A smaller step-size results in slower convergence but less misadjustment. It should also be noted that, if the step-size is too large,

10   then this iteration may diverge. This research presents no proofs nor performs any investigations into the range of step-size that guarantees convergence of this LENS gradient-search. Simulations suggest, however, that the general gradient-based convergence rule

$$\begin{array}{c} \text{Iteration} \\ \text{Convergence} \end{array} \iff \mu = \frac{\alpha}{E[\underline{X}^H \underline{X}]}, \quad 0 < \alpha < 1,$$

15   yields acceptable results provided that the parameter magnitudes, $|\hat{\beta}_{opt,i}[k]|^2$, $i = \{1, 2, \cdots, M-1\}$, are not too large. This research imposes the following modification upon the step-size in order to reduce the step size when the $|\hat{\beta}_{opt,i}[k]|^2$ are large:

$$\mu[k] = \frac{\alpha}{K E[\underline{X}^H \underline{X}]}, \quad 0 < \alpha < 1,$$
$$\text{with} \quad K = \max[1, \frac{\sum_{i=1}^{M-1} |\hat{\beta}_{opt,i}[k]|^2}{M-1}]. \qquad (6.10)$$

In this case, the parameter K is restricted to be greater than 1 to ensure that it only

20   shrinks and never enlarges $\mu[k]$.

6.3.2 Gradient LENS Processing

The most direct gradient-search modification to accommodate an unknown and changing input covariance would use the input stream, $\{\underline{X}^{(l)}\}$ to form $\hat{R}_{XX}^{(l)}$ and to apply a full gradient search to each $\hat{R}_{XX}^{(l)}$ to yield $\hat{\underline{\beta}}_{opt}^{(l)}$. Such a system would be inefficient, however, since gradient search iterations generally have long convergence times.

A simpler implementation may be realized by synchronizing the gradient-search iteration to the input observation iteration. Specifically, use $\hat{R}_{XX}^{(l)}$ to generate the current gradient $\nabla P[\underline{\beta}]^{(l)}$ [Equation 6.9] and substitute this into Equation 6.8 to update $\hat{\underline{\beta}}_{opt}^{(l)}$ into $\hat{\underline{\beta}}_{opt}^{(l+1)}$:

$$\hat{\underline{\beta}}_{opt}^{(l+1)} = \hat{\underline{\beta}}_{opt}^{(l)} - \mu^{(l)} \nabla P[\underline{\beta}]^{(l)}\big|_{\underline{\beta}=\hat{\underline{\beta}}_{opt}^{(l)}} . \tag{6.11}$$

For this implementation, the time-dependent step-size parameter defined in Equation 6.10 may be approximated by:

$$\mu^{(l)} = \frac{\alpha}{K \hat{\sigma}_{\underline{X}}^{2(l)}} \quad 0 < \alpha < 1,$$
$$\text{with} \quad K = \max[1, \frac{\sum_{i=1}^{M-1} |\hat{\beta}_{opt,i}^{(l)}|^2}{M-1}],$$

where $\hat{\sigma}_{\underline{X}}^{2(l)}$ is a running estimate of the total expected array input power, $E[\underline{X}^H \underline{X}]$.[1].

In general, each update step in Gradient LENS processing (or in any gradient-based adaptive algorithm) is small. Therefore, the adaptive Gradient LENS processing does not require a highly accurate estimate of the input covariance to yield acceptable results. The Gradient LENS processing estimates $\hat{R}_{XX}^{(l)}$ as the instantaneous input covaraince:

$$\hat{R}_{XX}^{(l)} = \underline{X}^{(l)}\underline{X}^{(l)H}. \qquad\qquad (6.12)$$

This reflects the covariance matrix estimate used by the traditional Frost Processing and Scaled Projection Algorithm gradient-based beamforming implementations [Section 2.3.3].

6.4 Computational Complexity

Figure 17 plots the number of real multiplications required per input snap shot, , $\underline{X}^{(l)}$, by Complete Update Relaxation, Partial Update Relaxation, and Gradient LENS S Processing.

Notably, the present invention separates the robustness constraining process from the beamforming power minimization, in contrast to prior art techniques which combine the robustness constraint into the beamforming power minimization. One of ordinary skill will recognize that the signal processing technique of the present invention has a wide variety of system applications.

Although the present invention has been shown and described with respect to several preferred embodiments thereof, various changes, omissions and additions to the form and detail thereof, may be made therein, without departing from the spirit and scope of the invention.

What is claimed is: